D
NA
…RNIA 93943

NAVAL POSTGRADUATE SCHOOL
Monterey, California

INVESTIGATION OF NON-LINEAR ESTIMATION OF NATURAL RESONANCES IN TARGET IDENTIFICATION

by

Choong Y. Chong

December 1983

Thesis Advisor: M. A. Morgan

Approved for public release; distribution unlimited.

T215125

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) Investigation of Non-linear Estimation of Natural Resonances in Target Identification	5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; December 1983
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Choong Y. Chong	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93943	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93943	12. REPORT DATE December 1983
	13. NUMBER OF PAGES 109
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report) Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Scattering Transient Response
The New Signal Model
Entire Function
Early Time Signal
Prony's Algorithm
Lease-Squares Method
Non-linear Optimization

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This investigation considers a non-linear technique for extracting natural resonances from transient electromagnetic scattering responses of radar targets. These natural resonances represent the complex poles of the target's transfer function in the Laplace transform s-plane. The advantage of their use in identification is their dependence only upon the

Block 20 Contd.

geometry and composition of the target and not upon the aspect and polarization of the incident signal.

Based on recent theoretical research, a new signal model has been developed for describing the form of the transient scattering response. This form precludes the optimal use of previous methods for natural resonance extraction based on Prony's algorithm. In this effort, a modified least-squares approach is taken which can accommodate the actual form of the transient signal. The algorithm is tested using simulated signals with various noise levels and conclusions are drawn concerning the viability of the method.

Approved for public release; distribution unlimited.

INVESTIGATION OF NON-LINEAR ESTIMATION OF NATURAL RESONANCES IN TARGET IDENTIFICATION

by

Choong Y. Chong
Lieutenant, Korean Navy
B.S., Seoul National University, 1980

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL

December 1983

ABSTRACT

This investigation considers a non-linear technique for extracting natural resonances from transient electromagnetic scattering responses of radar targets. These natural resonances represent the complex poles of the target's transfer function in the Laplace transform s-plane. The advantage of their use in identification is their dependence only upon the geometry and composition of the target and not upon the aspect and polarization of the incident signal.

Based on recent theoretical research, a new signal model has been developed for describing the form of the transient scattering response. This form precludes the optimal use of previous methods for natural resonance extraction based on Prony's algorithm. In this effort, a modified least-squares approach is taken which can accommodate the actual form of the transient signal. The algorithm is tested using simulated signals with various noise levels and conclusions are drawn concerning the viability of the method.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

This investigation considers a non-linear technique for extracting natural resonances from transient electromagnetic scattering responses of radar targets. These natural resonances represent the complex poles of the target's transfer function in the Laplace transform s-plane. The advantage of their use in identification is their dependence only upon the geometry and composition of the target and not upon the aspect and polarization of the incident signal.

Concepts and methodologies evolving from this idea have been developed based on the Singularity Expansion Method (SEM) by Baum [Ref. 1] in which the location of s-plane singularities characterizes a target. The direct extraction of poles (and residues) from the complex impulse response was initially attempted by Main and Moffat [Ref. 2]. These two early references spawned the idea of using natural resonances as the basis of non-cooperative target recognition (NCTR).

The application of the Householder orthogonalization method using a successive orthogonalization process and the eigenvalue method, that finds eigenvectors and eigenvalues, can determine the number and value of poles from the traditional transient signal model of a scatterer if there is a sufficiently low level of noise in the data [Ref. 3]. But both methods show poor quality of performance if we apply

them to the new signal model, even were there is no noise assumed in the signal data. This new signal model, which represents the electromagnetic scattering transient response of actual targets, was recently derived by Morgan [Ref. 4].

As considered in [Ref. 2], the relationship between a scattering target and the singularities being extracted from the natural complex resonance signal has been emphasized in the context of the number of poles and the accuracy of poles to ensure the unique relationship between an object and its image. Namely, the accuracy of pole extraction plays an important part in setting baselines to check consistency between a received signal and a set of singularities. Also, [Ref. 2] describes the modeling problem of a signal such that we express a transient response signal in terms of complex exponentials and the way it can be manipulated by Prony's algorithm. This signal model (termed here as the traditional model), was found not to represent the actual transient signal in the recent work by Davenport [Ref. 5]. In 1982, Manilha, in his work, commented that the new signal model might be partitioned into "early time signal" and "late time signal", each of which characterize the signal by the time instant at which the traditional model can be acceptable or not [Ref. 3]. This effort considers the development of a method that is capable of handling the "early time" whose time varying residues of exponentials do not become constant until the excitation of the incident field disappears. Such a signal

can not be adequately or accurately modeled by the traditional model.

It is the intention of this work to investigate a method that can handle the "early time signal" and show "Robustness" under relatively heavy noise pollution as well as improving the accuracy of the poles we are interested in evaluating. A non-linear parameter optimization approach, through the modified least-squares minimization method, has been tried. In particular, we have been interested in evaluating its performance in improving the accuracy of the poles extracted. The non-linear parameter optimization approach using iterations has contribued to improvement of accuracy and showed the possibility of application to the "early time signal" as was the final goal. We define the "new signal model" as a causal connection of the "early time signal" and "late time signal" throughout this paper. The new signal model will be presented in Chapter II, with both the description of the transient response mechanism and the way in which the signal model is constructed through the transient response mechanism. Chapter III briefly describes the problems associated with the traditional model, and with many of the current extraction methods in existence. Chapter IV is devoted to the description of the non-linear parameter optimization through a modified least-squares method and the way in which the algorithm is implemented. Chapter V presents test results of an attempt to use this approach for pole extraction and its performance characteristics. Chapter VI contains a summary

of the simulation results, the problems encountered, the potential problems expected, and the possibility of application of this method.

Computations were performed using the IBM 3033 system at the Naval Postgraduate School. The testing of concept feasibility was the major impetus behind this effort, with concerns for practical implementation in a "real-time" environment left for subsequent study. Considerations to the processing speed will not be addressed in this thesis. All the calculations were done in a double precision environment.

II. SCATTERING TRANSIENT RESPONSE SYSTEM

A. TRANSIENT RESPONSE MECHANISM

An electromagnetic wave incient upon a scattering body forces currents to be distributed such that Maxwell's equations and the corresponding boundary conditions are satisfied. These induced currents produce a scattered field that can be defined at any coordinate in free space if the expressions for those currents are made correctly. Changes in the angle of incidence, the incident wave shape, the polarization mode, and the target geometry and composition affect the results of the evaluation of the scattered transient response.

Basically, the singularities uniquely characterize a scatterer by their number and locations. The corresponding residues, which are interpreted as the weights of these poles, either can be viewed in the frequency domain or in the time domain.

As shown in [Ref. 3], equation (2.1) represents the form of the scattered signal impulse response of the target, in digital form

$$H(k) = H_{\emptyset}(kT) + \sum_{m=1}^{\infty} H_k \exp(kS_m T) \qquad (2.1)$$

where $H_{\emptyset}(t) = \emptyset$ for $t \geq T\emptyset$, T is the sampling interval and the infinite sum represents the natural resonance response with complex $S_m = \sigma_m + j\omega_m$ in the left-half plane. The poles constitute the parameters for NCTR of an object. The process

of extracting those parameters from the impulse response first collects the data points within a finite time period (time window).

The data collected is represented using equation (2.1) and equation (2.3) which partitions the signal into "early time" and "late time" components.

$$X(k) = \sum_{k=0}^{\infty} H(k) * \sum_{k=0}^{M-1} I(k) \tag{2.2}$$

$$X(k) = \underbrace{\sum_{k=0}^{T\emptyset-1} H(k) * \sum_{k=0}^{M-1} I(k)}_{\text{early time data}} + \underbrace{\sum_{k=T\emptyset}^{\infty} H(k) * \sum_{k=\emptyset}^{M-1} I(k)}_{\text{late time data}} \tag{2.3}$$

where $I(k)$ is the incident wave at $t = kT$

T is sampling interval in time.

We can write the equation (2.3) in the form of equation (2.4) or (2.5). Both of these expressions display the essence of the new signal model which will be discussed in the next section.

$$X(k) = \sum_{n=1}^{N} A_n (k) \exp(S_n k) \tag{2.4}$$

where $A_n(k) = A_n$ for $k \geq T\emptyset$

$$X(k) = E(k) + \sum_{n=1}^{N} A_n \exp(S_n k) \tag{2.5}$$

where $E(k) = \emptyset$ for $k \geq T\emptyset$

and A_n is constant for all k

TØ is the point at which the late time signal starts, so we consider that a time window might contain the data points ranging from the one before TØ to that after TØ. If we perform processing on a data window containing the data points which were from that early time region, we may lose some of the poles of natural resonance what are likely to be seen in the late time region. The data points after KØ presumably has all the information regarding poles, even though there are some extra poles that may be discriminated from the natural poles. These extra poles are introduced by the result of noise with relatively low SNR. Also, there is another risk to lose the natural poles having very high damping coefficients. There is no effective method yet found to compromise on having the point in which we can see all the poles with less extra poles under the higher SNR condition.

Meanwhile, the discrimination of E(k) appears to us as the quantity to be handled to give improvement in accuracy of the natural poles. The next section describes the new signal model based on these aspects and the transient response mechanism.

B. IMPULSE RESPONSE SIGNAL MODEL

The late time impulse response can be expressed as a summation of exponentially damped sinusoids as in equation (2.6).

$$X_a(k) = \sum_{n=1}^{N} A_n \exp(S_n k),\ k=\emptyset, T, 2T, \ldots, (M-1)T \qquad (2.6)$$

where $T = \Delta t$, and M is the number of sampling points. The signal model expressed in equation (2.6) is valid after the incident wave has completely illuminated a target and no forced response remains.

As the mechanism of a transient response system is strictly governed by the physics and orientation of the target and the attributes of the incident wave, we can write the equation (2.7) as an implicit form of equation (2.2), assuming we strike the target with a plane wave impulse.

$$X_b(k) = \sum_{k=\emptyset}^{\infty} H(k) \cdot \sum_{k=\emptyset}^{T\emptyset-1} [u(k) - u(k-T\emptyset)] \qquad (2.7)$$

Combining both equations (2.6) and (2.7), equation (2.9) would be sufficient if we assume N poles are present.

$$X(k) = X_a(k) + X_b(k)$$

$$X(k) = \sum_{k=\emptyset}^{\infty} H(k) \cdot \sum_{k=\emptyset}^{T\emptyset-1} [u(k) - u(k-T\emptyset)]$$

$$+ \; A_n \exp(S_n k), \; k=\emptyset, T, 2T, \ldots, (M-1)T \qquad (2.9)$$

We call the impulse response signal model, in the form of equation (2.9), an implicit form of the new signal model. A simpler form of equation (2.9) can be either of equations (2.1Ø) or (2.11).

$$X(k) = E(k) + \sum_{n=1}^{N} A_n \exp(S_n k) \qquad (2.1\emptyset)$$

$$X(k) = \sum_{n=1}^{N} A_n(k) \exp(S_n k) \qquad (2.11)$$

where $A_n(k) = A_n$, for $k \geq T\emptyset$

In equation (2.1), the residues are time-varying until the time at which the augmented function E(k) in equation (2.1Ø) becomes zero. Thereafter the $A_n(k) = A_n$ (constant).

To begin with equation (2.1Ø), it is necessary to define a new independent variable as a parameter whose time behavior is not predictable along the positive time axis. It is convenient to define this as a set of variables. By regarding them as independent, the equation (2.11) would be in the form of the equation (2.12) in the discrete digital data processing sense.

$$X(k) = E(k) + \sum_{n=1}^{N} A_n \exp(S_n k) \tag{2.12}$$

$$X(k) = \sum_{n=1}^{T\emptyset-1} e_n(k) + \sum_{n=1}^{N} A_n \exp\,(S_n k) \tag{2.13}$$

assuming $e_n(k) = E_n \delta(k-n)$

The equation (2.14) is expressed in terms of a positive counting sequence in k. As in the equation, we might evaluate the equation (2.13) if we know the exact number of poles. But we know that it is impossible because we are in the region of the early time. So we write the equation (2.14) as a time shifted version such as the equation that has meaning from the observability viewpoint.

$$X(k) = \sum_{n=1}^{T\emptyset-1} e_n(k) + \sum_{n=1}^{N} A_n \exp(S_n k) \tag{2.14}$$

where $k=T\emptyset+m, T\emptyset+m-1, \ldots, T\emptyset+1, T\emptyset, T\emptyset-1, \ldots, T\emptyset, \emptyset$

for $m \geq 0$

Here, the examples of the new signal model are presented in Appendix C, in the form of a decomposed signal. This synthetically generated signal is written in terms of $e_n(k)$'s and the sum of exponentials. In Chapter V, we present the decomposed signal from the synthetically generated data signal and that obtained by constructing the $E(k)$ as the results of the non-linear parameter optimization processing.

C. NOISE

White Gaussian noise is assumed throughout this work. By introducing the noise into the new impulse signal model, we call the following equation the modeling function of a transient response system and the complete and general form of the new signal model.

$$X(k) = N(k) + \sum_{n=1}^{\infty} e_n(k) + \sum_{n=1}^{N} A_n \exp(S_n k) \tag{2.15}$$

III. PROBLEMS WITH THE TRADITIONAL METHODS

In this chapter, a brief description of problems with the classic Prony's method, the Householder orthogonalization technique and the Eigenvalue method are provided. Each of these methods computes the location of poles in the s-plane but only the last two methods can estimate the number of poles prior to the discrimination of the poles.

A. PRONY's ALGORITHM

Equation (2.3) is interpreted as the following difference equation assigning the values of the real part of the signal at specified sampling points in its left side of the expression.

$$X(0) = \sum_{n=1}^{N} A_n$$

$$X(k) = \sum_{n=1}^{N} A_n Z_n^k \qquad (3.1)$$

$$\vdots$$

$$X((M-1)k) = \sum_{n=1}^{N} A_n Z_n^{(M-1)k}$$

And a polynomial equation (3.2) that has the same roots Z_n

$$\sum_{n=1}^{N} a_n Z^n = 0 \qquad (3.2)$$

can be combined with equation (3.1) to yield the Prony's difference equation in the form of equation (3.3)

$$\sum_{n=1}^{N} a_n X(n+k) = 0, \quad \text{for} \quad k=0,T,2T, \ldots , (M-1)\,T \qquad (3.3)$$

Then, we write equation (3.4) from equation (3.3) as

$$\sum_{n=1}^{N} a_n X(n+k) = \sum_{n=1}^{N-1} a_n X(n+k) + a_N x(n+k) = \emptyset \qquad (3.4)$$

Using matrix notation, the equation (3.3) may be written as equation (3.5) or (3.6).

$$\sum_{n=1}^{N-1} a_n X(n+k) = -a_N X(n+k) \qquad (3.5)$$

Let $a_N = 1$, then equation (3.5) becomes as

$$X_{N-1}\, A_{N-1} = x_N \qquad (3.6)$$

where X_{N-1} is N by N circulant matrix of sampled data

A_{N-1} is N by 1 Prony's coefficients matrix

x_N is N by 1 row matrix of sampled data set

$[X_{N}, _{N+1}, \ldots , X_{2N-1}]$

The Prony's coefficient a_n's are to be calculated by making use of the characteristics of the circulant matrix X.

$$A_{N-1} = (A_{N-1}\, X_{N-1})^{-1}\, A_{N-1}\, x_N \qquad (3.7)$$

As described in the above, this method can be applied after the number of poles are known to us, then the value of the S_n's are to be found from the equation (3.8).

$$S_n = LN(Z_n)/k, \text{ for } k \geq 2N \tag{3.8}$$

B. HOUSEHOLDER ORTHOGONALIZATION METHOD

A general expression of equation (3.2) may be represented by equation (3.9) assuming the number of poles are not known.

$$a_0 + a_1 Z^1 + \ldots + a_{N'} Z^{N'} = 0 \tag{3.9}$$

where N' is the unknown variable

Then, the equation (3.9) is to be satisfied under the condition that there are enough sampled data points $m \geq 2N$.

$$a_0 x_0 + a_1 x_1 + \ldots + a_{N'} x_{N'} = 0 \tag{3.10}$$

where the N by 1 sampled data matrix is defined as

$$x_i = [X(i),\ X(i+2),\ \ldots,\ X(M-N-i+1)]$$

Application of the successive orthogonalization through the Gram-Schmit process would produce the orthogonal vector set.

$$0 = \{o_0,\ o_1,\ \ldots,\ o_{N'}\}$$

$$\text{where } o_n = x_n - \sum_{i=0}^{n-1} \langle x_n, o_i \rangle\, o_i \tag{3.11}$$

If we set the o_0 to be 1, then the N by 1 x_n matrix is going to be in the form of equation (3.11).

$$x_n = o_n + \sum_{i=0}^{n-1} \langle x_n,\ o_i \rangle\, o_i \tag{3.12}$$

So that the equation (3.12) holds as $X=0M$. In the above, M is the 2-D matrix of multiplication factors whose diagonal

elements are all 1's. Here, an orthogonal vector o_1 makes the corresponding data set x_i to be orthogonal to all the previous vectors $x_\emptyset - x_{i-1}$. If we have had all the x vectors during i-1 times of successive orthogonalization process such that there is a non-zero orthogonal vector. We have to receive continuously the next data set x_{i+1} and check whether the orthogonal vector in the next step is zero or under a given threshold. If it vanishes, we can say that we have (i-1) poles. Our test against noise polluted data signal using the Householder method showed the inability of handling the early time signal as well as the data being heavily polluted.

C. EIGENVALUE METHOD

Again, from the equation (3.1∅), we can rewrite this equation as in the form of equation (3.14).

$$X_{N'} Z_{N'} = \emptyset \tag{3.14}$$

where $X_{N'} = [x_\emptyset | x_1 | \ . \ . \ . \ | x_{N'}]$

The eigenvector can be derived from the equation (3.15) by doing some mathematical manipulations of equation (3.14).

$$X^T{}_{N'} X_{N'} A_{N'} = X^T{}_{N'} X_{N'} E_{N'} = \emptyset$$

From this equation, we find the eigenvalues correspond to eigenvectors to see if there is any eigenvalue approaching zero. If there is one zero or under threshold, we also consider the number of poles as N'-1. Even both the

Householder orthogonalization method and the Eigenvalue method can provide the means of calculating the number of poles, their algorithm (tending to fit the data points to the traditional signal model) showed the lack of generality.

IV. NON-LINEAR PARAMETER OPTIMIZATION APPROACH

A. INTRODUCTION

In 1963, Marquardt suggested an algorithm for the least-squares estimation of non-linear parameters [Ref. 6] when highly accurate parameter values are required. As we can see in the signal model in equation (2.14), that new signal model has multiple parameters that are functions of time. Here, we rewrite that equation again in the form of equation (4.1).

$$X(k) = N(k) + \sum_{n=k}^{T\emptyset-1} e_n(k) + \sum_{n=1}^{N} A_n \exp(S_n k) \qquad (4.1)$$

where $e_n(k) = \emptyset$, for $k \geq T\emptyset$

We are not interested in the value of $e_n(k)$ itself, but include their estimations in order to contribute to the accuracy of S_n's and A_n's.

In other words, the extrapolation of the sum of exponentials, which is assumed to have all the poles, to points in the early time region may provide more accurate poles and residues simultaneously.

The major advantage is the fact that we can make use of the high SNR data signal in the early time region with the basic information of poles that were derived in the late time region.

We will be processing on a data window whose vector length is to be increased one by one by moving the first element of that vector toward the time-origin point. The optimization process evaluates the normal equation to find the local optimized set and finds the global optimized set as its final goal.

Let us define an ERROR function as in equation (4.3)

$$\mathrm{ERROR}\{A_n, S_n, E(k)\} = \sum_{n=1}^{N-1} [X'(k) - X(k)]^2 \tag{4.3}$$

where $X'(k)$ is a measured data point

To have minimized the least-squares error at every instant of measured time, the ERROR function has to be minimized by obtaining the global set of parameters being optimized. We also can write equation (4.3) in a more concrete manner as in equation (4.4).

$$\mathrm{ERROR}\ \{X\} = \sum_{k=\emptyset}^{M-1} [X'(k) - X(k)]^2 = \sum_{k=\emptyset}^{M-1} e(k)_n^2 \tag{4.4}$$

where $X = [E(k), A_n, S_n]$

Non-linear parameters such as $e_n(k)$ must be obtimized in a way such that the more accurate values of S_n have to be calculated as we increase the number of non-linear parameters e_n. Users may define the numbers of parameter e_n's that are at least equal to or greater than the number of data points in the early time region. Now it is our task to find out the optimized value of A_n, S_n, and e_n through optimization processing, either in the global sense as it is represented in the equation (4.5) or using the normal equation (4.6).

$$\partial ERROR(X)/\partial X = \emptyset \quad (4.5)$$

$$\partial e_n(k)/\partial X(k) = \emptyset \quad (4.6)$$

where $k=\emptyset, T, 2T \, . \, . \, . \, , (M-1)K$

$n=1,2, \, . \, . \, . \, , N$

With using the normal equation, as it is shown in the (4.6), a global set of non-linear set of parameters can be obtained through an iterative evaluation and fitting process. In our simulation work, the modified least-squares method was the basis of the equation (4.6).

B. THE MODIFIED LEAST-SQUARES METHOD

The error function is rewritten as in the expression of the equation (4.7).

$$ERROR\{k,x\} = X'(k) - X(k, x_n^k) \quad (4.7)$$

where $X'(k)$ is the k-th sampled data

$X(k, x)$ is the model function

x_n^k is a vector containing the k-th sampled

Let us define $x_n^\emptyset$ to be an initial estimated value of x, then a sequence of approximations to the optimized value is to be generated by the equation (4.8).

$$x^{m+1} = x^m - [a_m D_m + J_m^T J_m]^{-1} J_m^T \; ERROR(x^m) \quad (4.8)$$

where J_m is the numerical Jacobian matrix evaluated

m is iteration number of successive optimization

D_m is a diagonal matrix equal to the diagonal of

$J_m^T J_m$

a_m is a Marquardt parameter

The number of total iterations can be controlled by the threshold which is defined as in the equation (4.9).

$$d = x^{m+1} - x^m \tag{4.9}$$

V. TEST RESULTS AND PERFORMANCE EVALUATION

A. INTRODUCTION

In order to establish the ability of the program listed in Appendix B to extract the poles and residues correctly, three different simulated signals, each of which is polluted with infinite, 30 dB and 15 dB SNR, were created by the synthetic signal data generation routine. These 3 sets of signal data were chosen to span a range of possible situations.

In the context of the transient signals and additive stationary Gaussian noise being used, the definition of SNR is in terms of a ratio of energy quantities intergrated over the entire 20 nsec time window.

TABLE I

SIMULATED SIGNAL 1

RESIDUES	POLES (Nep. GHZ)
1.Ø + i1.Ø	-1.Ø + i1.Ø
1.Ø - i1.Ø	-1.Ø - i1.Ø

Simulated signal 2 consists of 2 sets of pairs of complex conjugate poles and residues, which is an extrapolation of the simulated signal 1, using its parameters. Simulated signal 3 has 3 sets of pole-residue pairs, extrapolating from signal 2.

There are 5 options in the signal generation program in choosing a particular function of E(k). In this simulation work, a trapezoidal wave form of E(k) was used with TØ=Ø .42 nsec. So that the unknown parameters that are to be augmented

at every processing step will be a maximum of 10 plus 4 times the number of poles at the final processing stage, when we have 512 data points within a 20 nsec time window.

TABLE II

SIMULATED SIGNAL 2

RESIDUES	POLES (Nep. GHZ)
1.Ø + i1.Ø	-1.Ø + i1.Ø
1.Ø - i1.Ø	-1.Ø - i1.Ø
Ø.5 + iØ.5	-2.Ø + i2.Ø
Ø.5 + iØ.5	-2.Ø - j2.Ø

TABLE III

SIMULATED SIGNAL 3

RESIDUES	POLES (Nep. GHZ)
1.Ø + i1.Ø	-1.Ø + i1.Ø
1.Ø - i1.Ø	-1.Ø - i1.Ø
Ø.5 + iØ.5	-2.Ø + i2.Ø
Ø.5 - iØ.5	-2.Ø - i2.Ø
Ø.25 + iØ.25	-3.Ø + i3.Ø
Ø.25 - iØ.25	-3.Ø - i3.Ø

The results of varying the number of additional data points from zero (all the data points are from late time region) to ten extra data points (10 extra points are from early time region are added to the points of late time region) are contained in Table IV through XII. These nine

cases correspond to the 3 synthetic signals, each having 3 additive noise levels. The 3 synthetic signals are plotted in Appendix C. In the next section, the accuracy of pole extractions, as indicated in the tables are displayed by way of graphical pole maps. In addition, reconstructed waveform obtained from the parameter extractions are compared graphically to the original waveforms.

TABLE IV

PARAMETER OPTIMIZATION FOR SIGNAL 1 (NOISE FREE)

B. RESULTS

```
TARGET   TYPE:TGT-1
WAVEFORM TYPE:HCNFTR
CONTACT  DATE:DEC 15
FILE     NAME:FILE002
NUMB. OF POLE:   2
```

TABLE OF RESIDUES AND POLES

PAIR #	RES-REAL	RES-IMAG	POLE-REAL	POLE-IMAG
1	1.00000000	1.00000000	-1.00000000	1.00000000
2	1.00000000	-1.00000000	-1.00000000	-1.00000000

RESULTS OF OPTIMIZATION WITH 0 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.99996610	1.00001592	-0.99994705	0.99998971
2	0.99996563	-1.00001375	-0.99994777	-0.99998971

RESULTS OF OPTIMIZATION WITH 2 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.00000251	0.99999667	-1.00000025	1.00000085
2	1.00000251	-0.99999667	-1.00000026	-1.00000085

RESULTS OF OPTIMIZATION WITH 4 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.00001308	1.00000329	-1.00001536	0.99999999
2	1.00001311	-1.00000330	-1.00001545	-1.00000002

RESULTS OF OPTIMIZATION WITH 6 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.99985213	1.00003040	-0.99979429	0.99998208
2	0.99985210	-1.00003047	-0.99979441	-0.99998213

RESULTS OF OPTIMIZATION WITH 8 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.00000036	1.00001011	-1.00000472	0.99999818
2	1.00000026	-1.00001041	-1.00000540	-0.99999838

RESULTS OF OPTIMIZATION WITH 10 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.00000457	0.99999062	-0.99999674	1.00000165
2	1.00000453	-0.99999067	-0.99999735	-1.00000168

TABLE V

PARAMETER OPTIMIZATION FOR SIGNAL 1
(SNR=30 dB)

```
TARGET     TYPE:TGT-1
WAVEFORM   TYPE:FCN30TR
CONTACT    DATE:DEC 15
FILE       NAME:FILE003
NUMB. OF   POLE:   2
```

TABLE OF RESIDUES AND POLES

PAIR #	RES-REAL	RES-IMAG	POLE-REAL	POLE-IMAG
1	1.00000000	1.00000000	-1.00000000	1.00000000
2	1.00000000	-1.00000000	-1.00000000	-1.00000000

RESULTS OF OPTIMIZATION WITH 0 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.01006372	0.92346346	-0.77676493	1.00870076
2	1.03315188	-1.10738236	-1.30247623	0.99197436

RESULTS OF OPTIMIZATION WITH 2 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.01070336	1.00360284	-1.00951167	0.99979304
2	1.01070339	-1.00360275	-1.00951163	0.99979306

RESULTS OF OPTIMIZATION WITH 4 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.01066217	1.00367655	-1.00953849	0.99977751
2	1.01066632	-1.00367409	-1.00953782	-0.99977735

RESULTS OF OPTIMIZATION WITH 6 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.01059740	1.00357123	-1.00930702	0.99978580
2	1.01059679	-1.00357126	-1.00930751	0.99978574

RESULTS OF OPTIMIZATION WITH 8 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.01067940	1.00362571	-1.00950587	0.99978686
2	1.01068033	-1.00362636	-1.00950588	-0.99978699

RESULTS OF OPTIMIZATION WITH 10 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.01069621	1.00360379	-1.00950576	0.99979175
2	1.01069609	-1.00360399	-1.00950630	-0.99979177

TABLE VI

PARAMETER OPTIMIZATION FOR SIGNAL 1
(SNR=15 dB)

TARGET TYPE:TGT-1
WAVEFORM TYPE:HDN15TR
CONTACT DATE:DEC 20
FILE NUME:FILE004
NUMB. OF POLE: 2

TABLE OF RESIDUES AND POLES

PAIR #	RES-REAL	RES-IMAG	POLE-REAL	POLE-IMAG
1	1.00000000	1.00000000	-1.00000000	1.00000000
2	1.00000000	-1.00000000	-1.00000000	-1.00000000

RESULTS OF OPTIMIZATION WITH 0 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.56047982	1.39923892	-0.57890084	0.98886304
2	1.59727710	-0.80336947	-1.57013329	-0.91419374

RESULTS OF OPTIMIZATION WITH 2 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.06057099	1.02070489	-1.05306349	0.99869879
2	1.06057923	-1.02071795	-1.05306276	-0.99869878

RESULTS OF OPTIMIZATION WITH 4 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.06053386	1.02072970	-1.05304404	0.99868990
2	1.06052818	-1.02073354	-1.05304091	-0.99869088

RESULTS OF OPTIMIZATION WITH 6 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.06051499	1.02073547	-1.05295266	0.99873653
2	1.06054674	-1.02072794	-1.05313212	-0.99864419

RESULTS OF OPTIMIZATION WITH 8 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.02449720	1.09889916	-0.66485231	1.01775339
2	1.07048258	-1.10030318	-1.63907420	-0.93004124

RESULTS OF OPTIMIZATION WITH 10 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.03166928	1.11984550	-1.56675813	0.92761091
2	1.03351816	-1.07048377	-0.68289831	-1.02160947

TABLE VII

PARAMETER OPTIMIZATION FOR SIGNAL 2
(NOISE FREE)

```
TARGET    TYPE:TGT-2
WAVEFORM  TYPE:HDNFTR
CONTACT   DATE:DEC 15
FILE      NUME:FILE002
NUMB. OF  POLE:   4
```

TABLE OF RESIDUES AND POLES

PAIR #	RES-REAL	RES-IMAG	POLE-REAL	POLE-IMAG
1	1.00000000	1.00000000	-1.00000000	1.00000000
2	1.00000000	-1.00000000	-1.00000000	-1.00000000
3	0.50000000	0.50000000	-2.00000000	2.00000000
4	0.50000000	-0.50000000	-2.00000000	-2.00000000

RESULTS OF OPTIMIZATION WITH 0 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.00000054	0.99998637	-1.00000054	1.00000162
2	1.00000019	-0.99998792	-0.99999823	-1.00000180
3	0.49993635	0.50005618	-2.00005518	1.99995365
4	0.49995131	-0.50004431	-2.00024952	-1.99994115

RESULTS OF OPTIMIZATION WITH 2 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.99906513	0.99549512	-0.99792594	0.99998185
2	0.99906405	-0.99549467	-0.99792606	-0.99998794
3	0.47735342	0.48850313	-1.95118748	1.99896225
4	0.47735838	-0.48850442	-1.95306779	-1.99850918

RESULTS OF OPTIMIZATION WITH 4 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.99906256	0.99549224	-0.99793418	0.99998541
2	0.99906321	-0.99549228	-0.99791438	-0.99998413
3	0.47737187	0.48848088	-1.95232122	1.99855642
4	0.47736975	-0.48847526	-1.95190026	-1.99893520

RESULTS OF OPTIMIZATION WITH 6 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.99906743	0.99549467	-0.99793003	0.99998501
2	0.99906717	-0.99549457	-0.99792354	-0.99998468
3	0.47739612	0.48847662	-1.95179495	1.99874576
4	0.47739443	-0.48847686	-1.95248987	-1.99875751

RESULTS OF OPTIMIZATION WITH 8 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.01514809	0.98882407	-1.12954971	1.01650809
2	1.00949031	-0.98873294	-0.88470066	-0.99067696
3	0.78175371	0.36507609	-2.03189450	2.04744671
4	0.14319333	-0.56814961	-1.95892411	-1.92582259

RESULTS OF OPTIMIZATION WITH 10 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.02706396	0.98982102	-1.16328021	1.01920699
2	1.02053154	-0.99088746	-0.86815870	-0.99077552
3	0.34656670	0.59967646	-1.95544269	1.95297875
4	0.61555028	-0.37246541	-2.00784742	-2.04312205

TABLE VIII

PARAMETER OPTIMIZATION FOR SIGNAL 2
(SNR=30 dB)

TARGET TYPE:TGT-2
WAVEFORM TYPE:HDN30TR
CONTACT DATE:DEC 15
FILE NUMB:FILE003
NUMB. OF POLE: 4

TABLE OF RESIDUES AND POLES

PAIR #	RES-REAL	RES-IMAG	POLE-REAL	POLE-IMAG
1	1.00000000	1.00000000	-1.00000000	1.00000000
2	1.00000000	-1.00000000	-1.00000000	-1.00000000
3	0.50000000	0.50000000	-2.00000000	2.00000000
4	0.50000000	-0.50000000	-2.00000000	-2.00000000

RESULTS OF OPTIMIZATION WITH 0 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.99412945	0.99056805	-0.92918755	0.99302604
2	0.99548280	-0.99069937	-1.06493363	-1.00865606
3	0.00000010	0.56023544	-1.96891695	1.90034350
4	0.87406384	-0.32304349	-2.04566871	-2.05396332

RESULTS OF OPTIMIZATION WITH 2 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.99906513	0.99549512	-0.99792594	0.99998185
2	0.99906405	-0.99549467	-0.99792606	-0.99998794
3	0.47735342	0.48850313	-1.95118748	1.99896225
4	0.47735838	-0.48850442	-1.95306779	-1.99850918

RESULTS OF OPTIMIZATION WITH 4 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.99906256	0.99549224	-0.99793418	0.99998541
2	0.99906321	-0.99549228	-0.99791438	-0.99998413
3	0.47737187	0.48848088	-1.95232122	1.99855642
4	0.47736977	-0.48847526	-1.95190026	-1.99893520

RESULTS OF OPTIMIZATION WITH 6 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.99906743	0.99549467	-0.99793003	0.99998501
2	0.99906717	-0.99549457	-0.99792354	-0.99998468
3	0.47739612	0.48847662	-1.95179495	1.99874576
4	0.47739443	-0.48847686	-1.95248987	-1.99875751

RESULTS OF OPTIMIZATION WITH 8 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.01514809	0.98882407	-1.12954971	1.01650809
2	1.00949031	-0.98875294	-0.88470066	-0.99067696
3	0.78175371	0.36507609	-2.03189450	2.04744671
4	0.14319333	-0.56814961	-1.95892411	-1.92582259

RESULTS OF OPTIMIZATION WITH 10 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.02706396	0.98982102	-1.16328021	1.01920699
2	1.02053154	-0.99088746	-0.86815870	-0.99077552
3	0.34656670	0.59967646	-1.95544269	1.95297875
4	0.61555028	-0.37246541	-2.00784742	-2.04312205

TABLE IX

PARAMETER OPTIMIZATION FOR SIGNAL 2
(SNR=15 dB)

TARGET TYPE:TGT-2
WAVEFORM TYPE:HDN15TR
CONTACT DATE:DEC 15
FILE NUME:FILE004
NUMB. OF POLE: 4

TABLE OF RESIDUES AND POLES

PAIR #	RES-REAL	RES-IMAG	POLE-REAL	POLE-IMAG
1	1.00000000	1.00000000	-1.00000000	1.00000000
2	1.00000000	-1.00000000	-1.00000000	-1.00000000
3	0.50000000	0.50000000	-2.00000000	2.00000000
4	0.50000000	-0.50000000	-2.00000000	-2.00000000

RESULTS OF OPTIMIZATION WITH 0 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.96091205	0.96731225	-0.97658042	0.99822946
2	0.96092603	-0.96731915	-0.97544458	-0.99811586
3	0.52307777	0.22152280	-1.54313287	2.10614539
4	0.00000010	-0.42428522	-1.38794994	-1.83372419

RESULTS OF OPTIMIZATION WITH 2 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.96511311	0.96627888	-0.99390301	1.00027364
2	0.96526467	-0.96632064	-0.95966500	-0.99679353
3	0.00000010	0.49329107	-1.46905167	1.85021464
4	0.55450498	-0.13648901	-1.66572130	-2.12454838

RESULTS OF OPTIMIZATION WITH 4 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.96128516	0.96679315	-0.98935769	0.99965070
2	0.96144733	-0.96682559	-0.96226170	-0.99687448
3	0.00000010	0.54101355	-1.50069226	1.85232329
4	0.50251383	-0.09527370	-1.60980060	-2.13416731

RESULTS OF OPTIMIZATION WITH 6 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.06294479	0.93763540	-1.25949424	1.04372728
2	1.01484084	-0.94095523	-0.77277970	-0.98546978
3	0.62674427	0.16116490	-1.75036751	2.11201876
4	0.00000010	-0.51970056	-1.59036824	-1.84519866

RESULTS OF OPTIMIZATION WITH 8 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.96304511	0.96782052	-0.98449408	0.99905219
2	0.96312358	-0.96782270	-0.96940893	-0.99752145
3	0.00000010	0.39719908	-1.36627993	1.83147922
4	0.54632361	-0.25136131	-1.56644065	-2.10011869

RESULTS OF OPTIMIZATION WITH 10 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.00971067	0.96985420	-1.20987398	1.02332284
2	1.00285344	-0.97024550	-0.81824047	-0.98822709
3	0.57607620	0.19510021	-1.66621998	2.10722391
4	0.00000010	-0.50028280	-1.53834141	-1.84201571

TABLE X

PARAMETER OPTIMIZATION FOR SIGNAL 3
(NOISE FREE)

```
TARGET    TYPE:TGT-3
WAVEFORM  TYPE:HONFTR
CONTACT   DATE:DEC 15
FILE      NUMB:FILE004
NUMB. OF  POLE:   6
```

TABLE OF RESIDUES AND POLES

PAIR #	RES-REAL	RES-IMAG	POLE-REAL	POLE-IMAG
1	1.00000000	1.00000000	-1.00000000	1.00000000
2	1.00000000	-1.00000000	-1.00000000	-1.00000000
3	0.50000000	0.50000000	-2.00000000	2.00000000
4	0.50000000	-0.50000000	-2.00000000	-2.00000000
5	0.25000000	0.25000000	-3.00000000	3.00000000
6	0.25000000	-0.25000000	-3.00000000	-3.00000000

RESULTS OF OPTIMIZATION WITH 0 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.00000029	0.99999951	-1.00000013	1.00000000
2	0.99999965	-1.00000047	-0.99999985	-0.99999999
3	0.50000021	0.49999977	-1.99999733	1.99999965
4	0.49999957	-0.50000051	-2.00000290	-2.00000029
5	0.25000049	0.25000087	-2.99876241	2.99999566
6	0.24999768	-0.24999958	-3.00125324	-2.99999619

RESULTS OF OPTIMIZATION WITH 2 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.00000033	1.00000006	-0.99999640	0.99999972
2	0.99999970	-0.99999990	-1.00000359	-1.00000028
3	0.49999897	0.49999976	-2.00000082	1.99999976
4	0.50000019	-0.49999970	-1.99999827	-2.00000017
5	0.24999312	0.25000063	-2.99929864	2.99995040
6	0.25000768	-0.24999482	-3.00067868	-3.00005033

RESULTS OF OPTIMIZATION WITH 4 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.00000031	1.00000004	-0.99999929	0.99999999
2	1.00000008	-0.99999999	-1.00000084	-1.00000003
3	0.50000029	0.49999878	-2.00000158	2.00000057
4	0.50000073	-0.49999913	-1.99999748	-1.99999986
5	0.24999713	0.25000137	-3.00006822	3.00000834
6	0.25000306	-0.24999453	-2.99992441	-2.99999365

RESULTS OF OPTIMIZATION WITH 6 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.99999967	1.00000006	-0.99999978	0.99999979
2	0.99999949	-1.00000005	-1.00000003	-1.00000016
3	0.50000350	0.49999519	-2.00000411	2.00000118
4	0.50000357	-0.49999515	-2.00000565	-2.00000116
5	0.25002351	0.25003725	-3.01254931	2.99994026
6	0.24999017	-0.25002880	-2.98710887	-2.99987758

RESULTS OF OPTIMIZATION WITH 8 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.00000000	0.99999980	-0.99999966	1.00000026
2	1.00000001	-1.00000010	-1.00000031	-0.99999975
3	0.50000063	0.49995686	-2.00000881	1.99969134
4	0.49999755	-0.50003918	-1.99998120	-2.00030891
5	0.25086787	0.25234001	-3.00656519	3.00052848
6	0.24913768	-0.24765392	-2.99334543	-2.99547896

RESULTS OF OPTIMIZATION WITH 10 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	1.00000004	0.99999995	-1.00000147	1.00000006
2	0.99999994	-1.00000008	-0.99999854	-0.99999994
3	0.50000006	0.50000023	-2.00001924	1.99999846
4	0.50000019	-0.50000011	-1.99998131	-2.00000154
5	0.25000013	0.25000022	-2.99992168	2.99997583
6	0.25000024	-0.25000077	-3.00008161	-3.00002382

TABLE XI

PARAMETER OPTIMIZATION FOR SIGNAL 3
(SNR=30 dB)

TARGET TYPE:TGT-3
WAVEFORM TYPE:HDN30TR
CONTACT DATE:DEC 15
FILE NUMB:FILE004
NUMB. OF POLE: 6

TABLE OF RESIDUES AND POLES

PAIR #	RES-REAL	RES-IMAG	POLE-REAL	POLE-IMAG
1	1.00000000	1.00000000	-1.00000000	1.00000000
2	1.00000000	-1.00000000	-1.00000000	-1.00000000
3	0.50000000	0.50000000	-2.00000000	2.00000000
4	0.50000000	-0.50000000	-2.00000000	-2.00000000
5	0.25000000	0.25000000	-3.00000000	3.00000000
6	0.25000000	-0.25000000	-3.00000000	-3.00000000

RESULTS OF OPTIMIZATION WITH 0 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.99145389	0.98947089	-0.99318094	1.00001443
2	0.99145320	-0.98947086	-0.99310558	-1.00001237
3	0.42807658	0.52733218	-1.99204259	1.97341708
4	0.42846454	-0.52679082	-1.92072086	-1.99797545
5	0.11656813	0.23647261	-2.27124794	2.83611689
6	0.12939770	-0.21591933	-2.24119798	-3.06874116

RESULTS OF OPTIMIZATION WITH 2 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.99147340	0.98938441	-0.99323692	1.00002118
2	0.99147541	-0.98938631	-0.99299518	-1.00001483
3	0.42927584	0.52783363	-1.96123606	1.86208680
4	0.42938721	-0.52783004	-1.96042724	-1.98830969
5	0.13113198	0.20433430	-2.22870752	3.07350223
6	0.10954599	-0.24676563	-2.28661642	-2.83356831

RESULTS OF OPTIMIZATION WITH 4 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.99149793	0.98938042	-0.99300311	1.00001643
2	0.99149653	-0.98937734	-0.99323733	-1.00002228
3	0.42898254	0.52744729	-1.95708423	1.98692809
4	0.42897978	-0.52746018	-1.96340662	-1.98537735
5	0.13728153	0.20601283	-2.24720729	3.07453032
6	0.10518882	-0.24183713	-2.27543055	-2.83274078

RESULTS OF OPTIMIZATION WITH 6 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.99181508	0.98944282	-0.99314260	1.00003060
2	0.99181556	-0.98944354	-0.99332478	-1.00003549
3	0.43071414	0.52480665	-1.97634267	1.98151201
4	0.43071211	-0.52479742	-1.93934530	-1.99173878
5	0.15055056	0.25270674	-2.28536097	2.85503619
6	0.10533241	-0.20048107	-2.19638655	-3.07720611

RESULTS OF OPTIMIZATION WITH 8 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.98171120	0.98981745	-0.99074872	0.99947232
2	0.98171187	-0.98981744	-0.99060276	-0.99949877
3	0.30805732	0.49840155	-1.88871352	1.87285997
4	0.31532825	-0.51226642	-1.56132982	-2.02127811
5	0.11572596	0.13443903	-1.99741621	3.12706630
6	0.07739011	-0.19217344	-1.89148981	-2.85558132

RESULTS OF OPTIMIZATION WITH 10 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.99337417	0.99015792	-0.99418591	1.00008890
2	0.99337150	-0.99015803	-0.99369290	-1.00007652
3	0.44239942	0.52004030	-1.95770753	1.99113003
4	0.44239753	-0.52004768	-1.96547995	-1.98746271
5	0.15478007	0.23652063	-2.41828246	2.87099551
6	0.15377770	-0.23805068	-2.48497797	-3.06515871

TABLE XII

PARAMETER OPTIMIZATION FOR SIGNAL 3
(SNR=15 dB)

TARGET TYPE:TGT-3
WAVEFORM TYPE:FCN15TR
CONTACT DATE:DEC 15
FILE NUMB:FILE004
NUMB. OF POLE: 6

TABLE OF RESIDUES AND POLES

PAIR #	RES-REAL	RES-IMAG	POLE-REAL	POLE-IMAG
1	1.00000000	1.00000000	-1.00000000	1.00000000
2	1.00000000	-1.00000000	-1.00000000	-1.00000000
3	0.50000000	0.50000000	-2.00000000	2.00000000
4	0.50000000	-0.50000000	-2.00000000	-2.00000000
5	0.25000000	0.25000000	-3.00000000	3.00000000
6	0.25000000	-0.25000000	-3.00000000	-3.00000000

RESULTS OF OPTIMIZATION WITH 0 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.96625765	0.94693282	-0.96776691	1.00067712
2	0.96625659	0.94693466	-0.96776960	-1.00067721
3	0.25778800	0.55401616	-1.81344349	1.95442546
4	0.25778439	0.55401729	-1.81338128	-1.95442985
5	0.00000010	0.30602226	-2.40607412	2.86369297
6	0.00000010	-0.30602324	-2.40740413	-2.86372996

RESULTS OF OPTIMIZATION WITH 2 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.97003027	0.94583414	-0.96792409	1.00081726
2	0.97002941	-0.94583770	-0.96793108	-1.00081754
3	0.25168247	0.53580191	-1.78711241	1.95425944
4	0.25167543	-0.53579888	-1.78713746	-1.95423795
5	0.00000010	0.26710278	-2.24090416	2.86059552
6	0.00000010	-0.26681661	-2.24551510	-2.87417393

RESULTS OF OPTIMIZATION WITH 4 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.97144662	0.94762010	-0.96904247	1.00080514
2	0.97144593	-0.94762088	-0.96905146	-1.00080547
3	0.25997577	0.54275341	-1.80291116	1.95565286
4	0.25997714	-0.54275353	-1.80292514	-1.95564222
5	0.00000010	0.27664033	-2.30082557	2.86676095
6	0.00000010	-0.27744624	-2.27393152	-2.86430662

RESULTS OF OPTIMIZATION WITH 6 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.97074646	0.94593786	-0.96820496	1.00084896
2	0.97074740	-0.94593893	-0.96820018	-1.00084921
3	0.25317420	0.53401680	-1.78619717	1.95473602
4	0.25317481	-0.53401801	-1.78618621	-1.95473121
5	0.00000010	0.26390985	-2.23469098	2.86848662
6	0.00000010	-0.26397500	-2.23254981	-2.86625171

RESULTS OF OPTIMIZATION WITH 8 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.97233004	0.94707234	-0.96927693	1.00087271
2	0.97233259	-0.94707302	-0.96903353	-1.00086402
3	0.24416111	0.53457339	-1.78489285	1.95562845
4	0.24415737	-0.53465427	-1.79548721	-1.94953438
5	0.00000010	0.20134605	-1.70725522	2.99741937
6	0.00000010	-0.21428806	-1.61287861	-2.76554177

RESULTS OF OPTIMIZATION WITH 10 EXTRA DATA POINTS

PAIR #	RES.-REAL	RES.-IMAG	POLE-REAL	POLE-IMAG
1	0.97265829	0.94587240	-0.97006813	1.00100830
2	0.97266290	-0.94587709	-0.96753425	1.00089816
3	0.24466698	0.52883260	-1.77855047	1.95554555
4	0.24466804	-0.52886614	-1.78467362	1.95086597
5	0.00000010	0.25592515	-1.69220053	2.78203752
6	0.00000010	-0.15339728	-1.60711291	-3.01800673

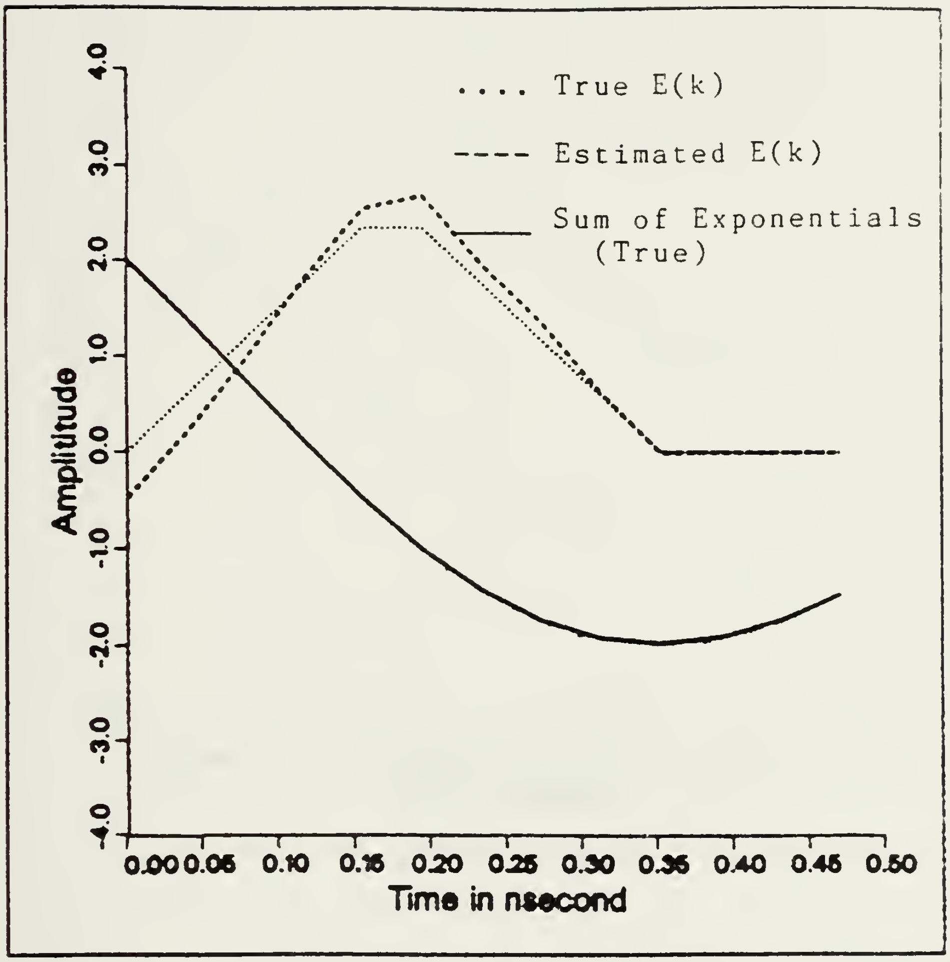

Figure 5.1. Estimated E(k) for Signal 1 of 15 dB SNR. (32 data points from late time region).

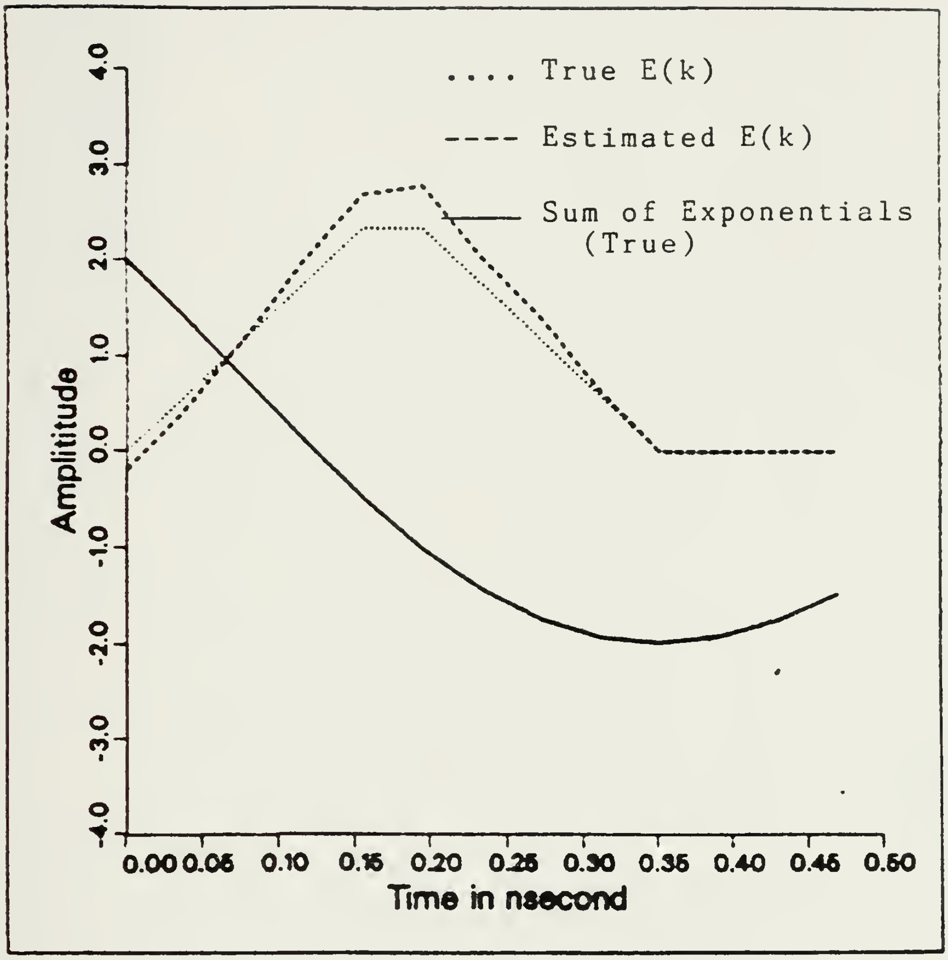

Figure 5.2. Estimated E(k) for Signal 1 of 15 dB SNR. (48 data points from late time region).

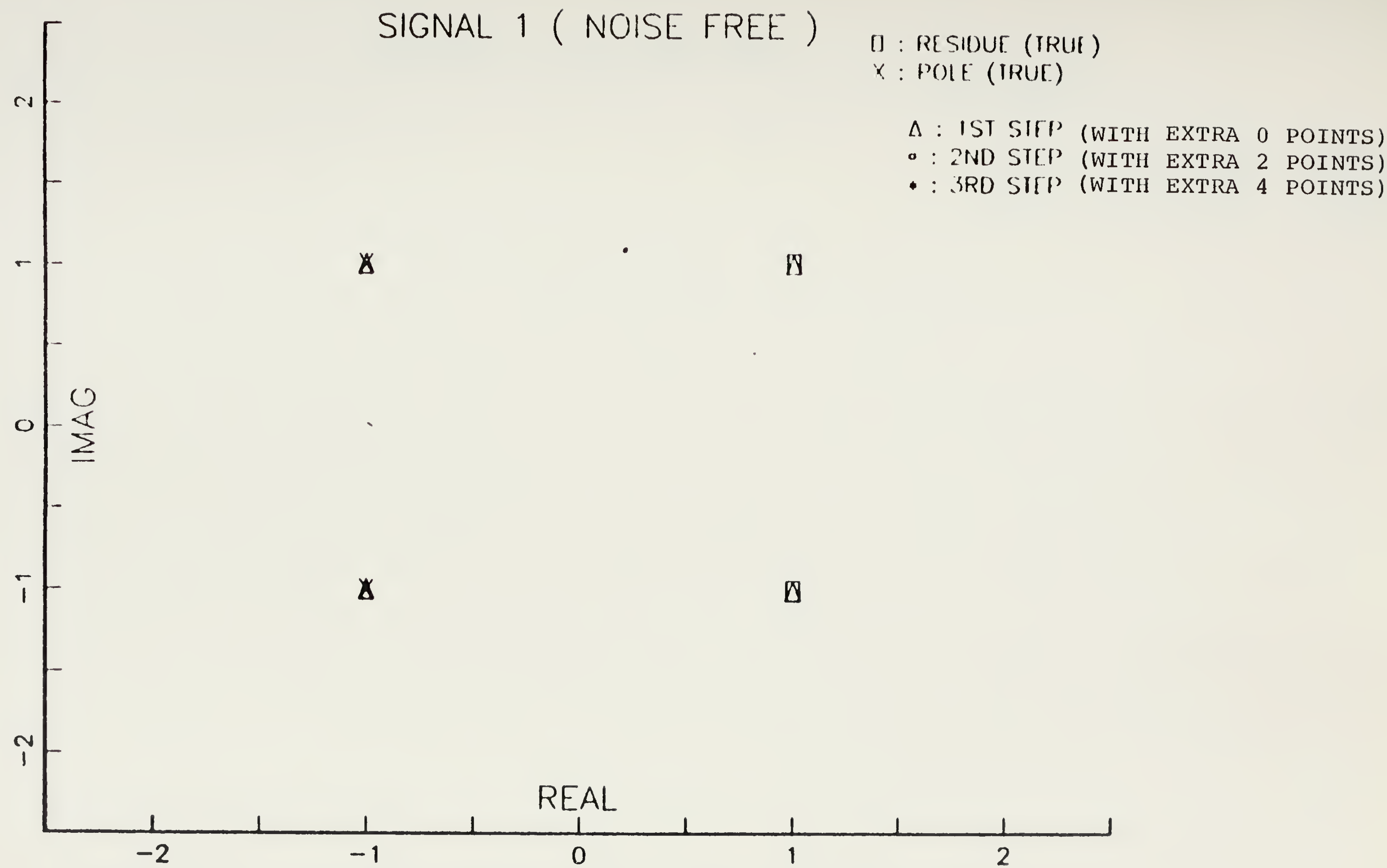

Figure 5.3. Pole and Residue Plot for Signal 1 of Noise Free.

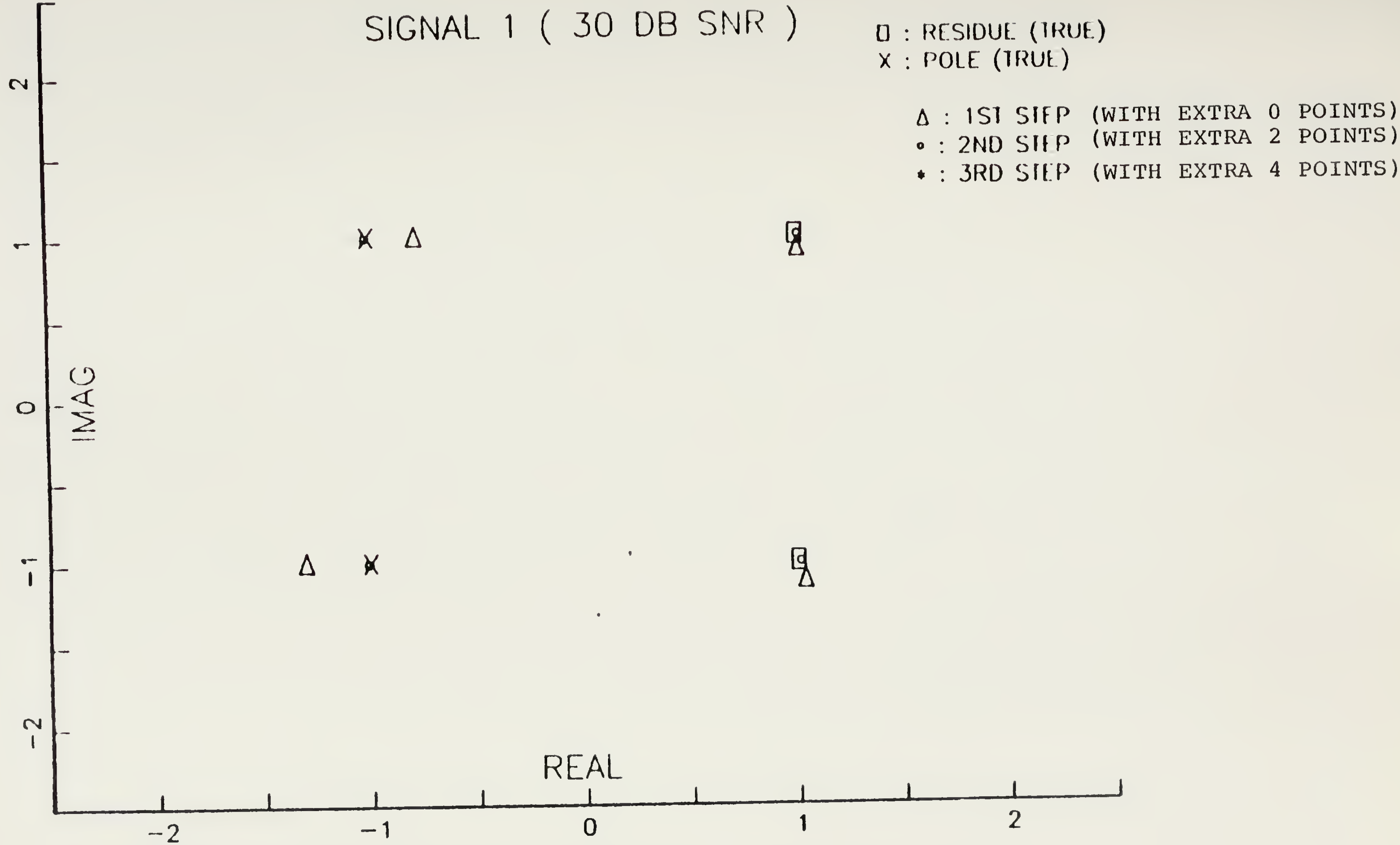

Figure 5.4. Pole and Residue Plot for Signal 1 of 30 dB SNR.

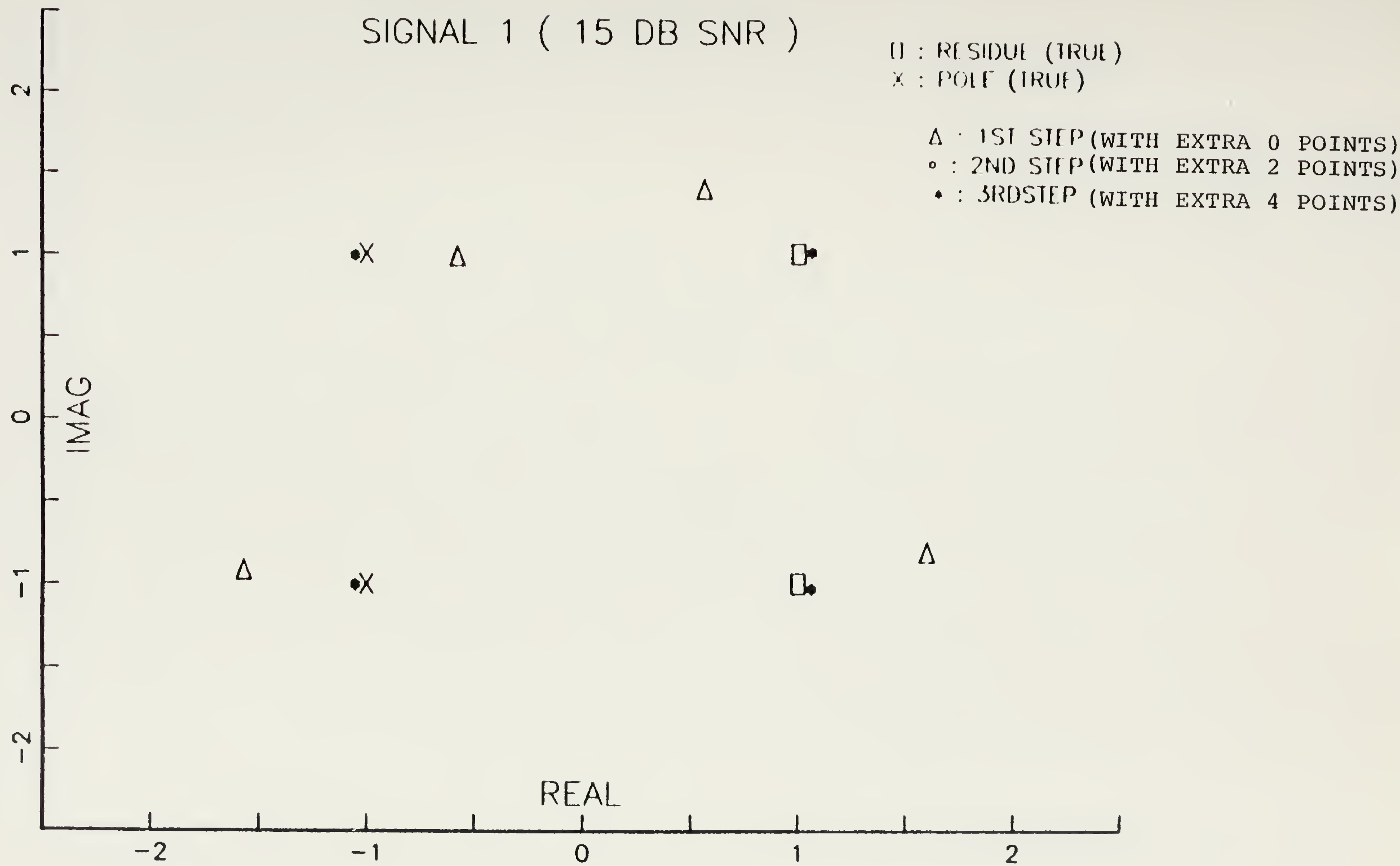

Figure 5.5. Pole and Residue Plot for Signal 1 of 15 dB SNR.

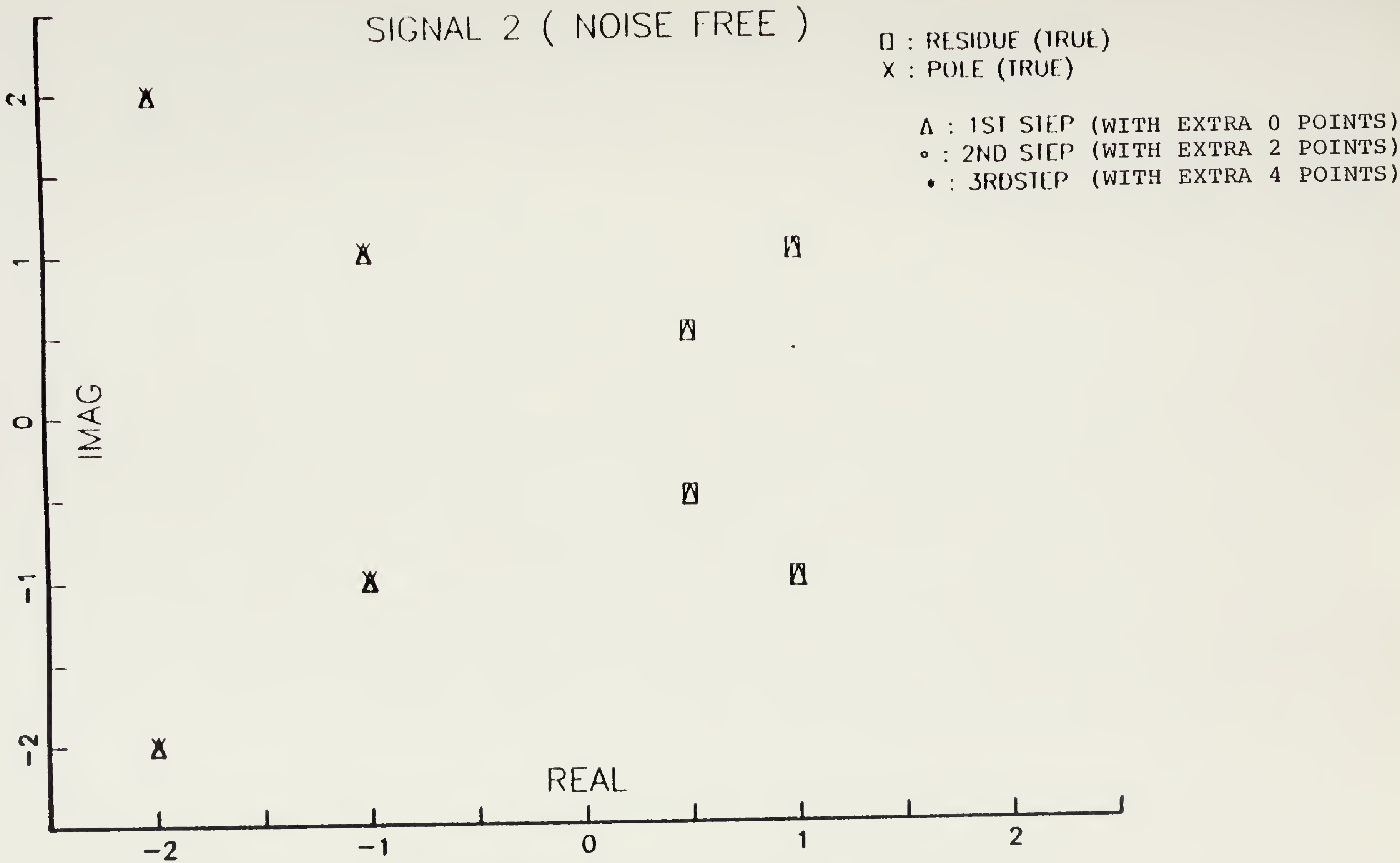

Figure 5.6. Pole and Residue Plot for Signal 2 of Noise Free.

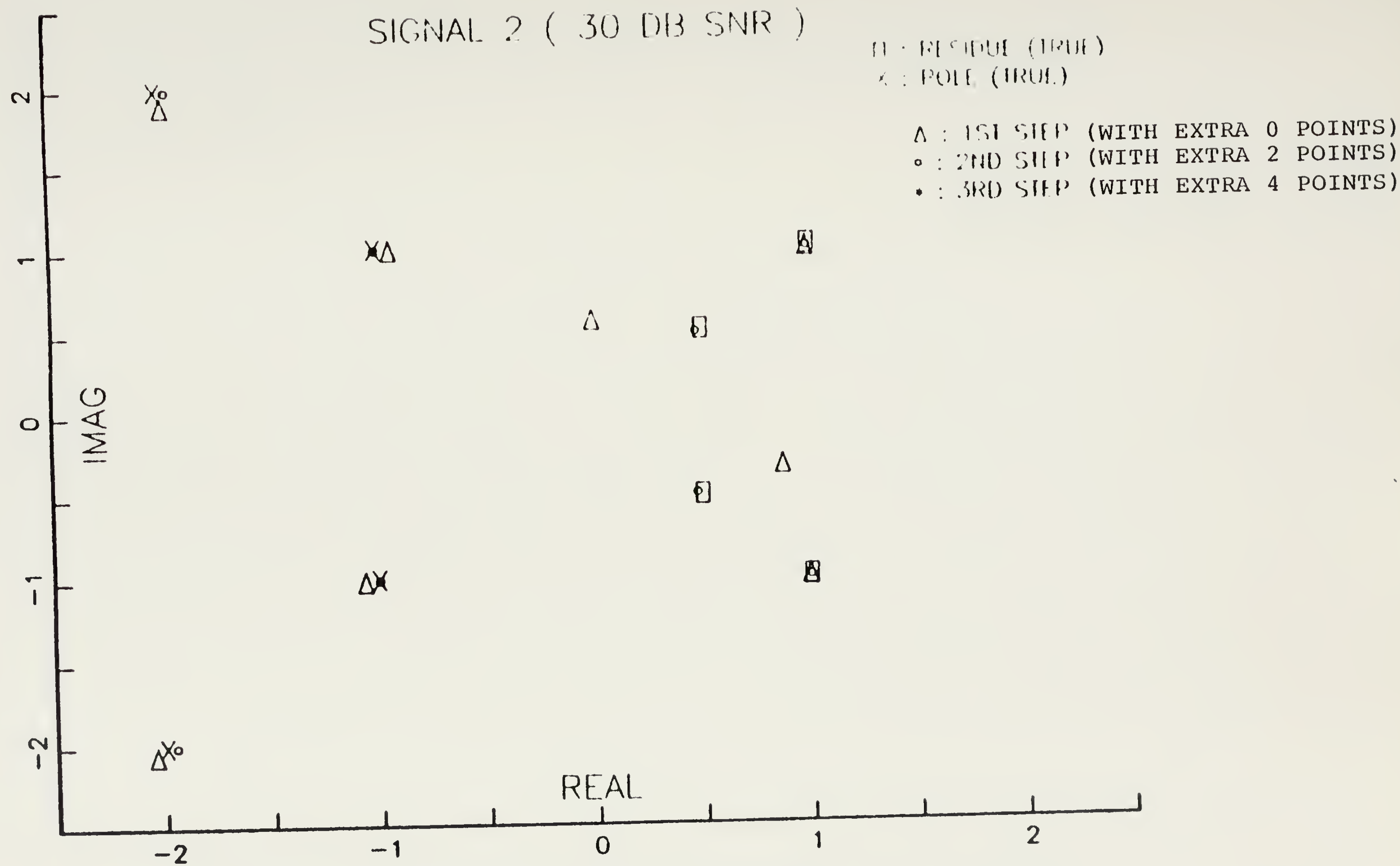

Figure 5.7. Pole and Residue Plot for Signal 2 of 30 dB SNR.

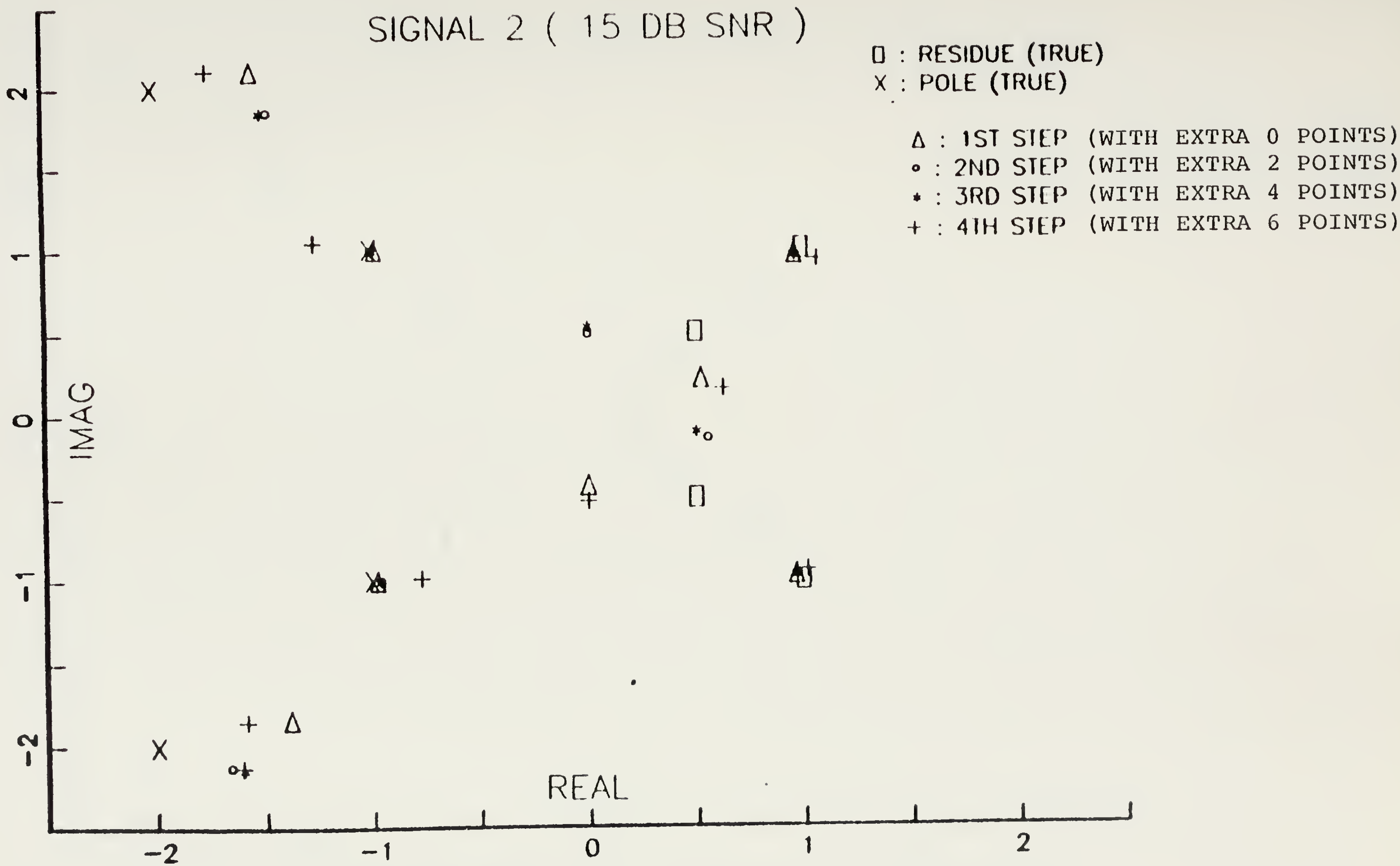

Figure 5.8. Pole and Residue Plot for Signal 2 of 15 dB SNR.

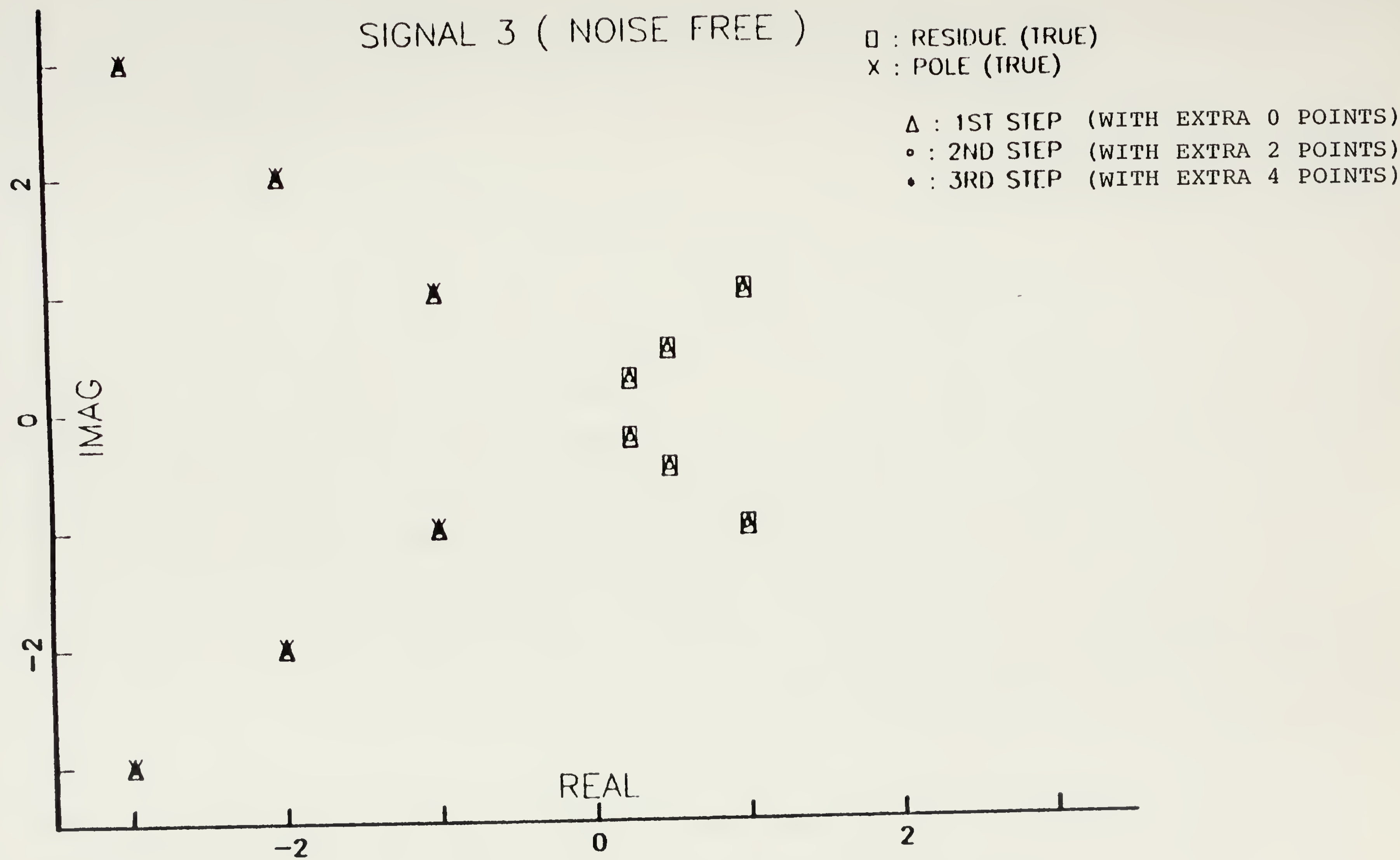

Figure 5.9. Pole and Residue Plot for Signal 3 of Noise Free.

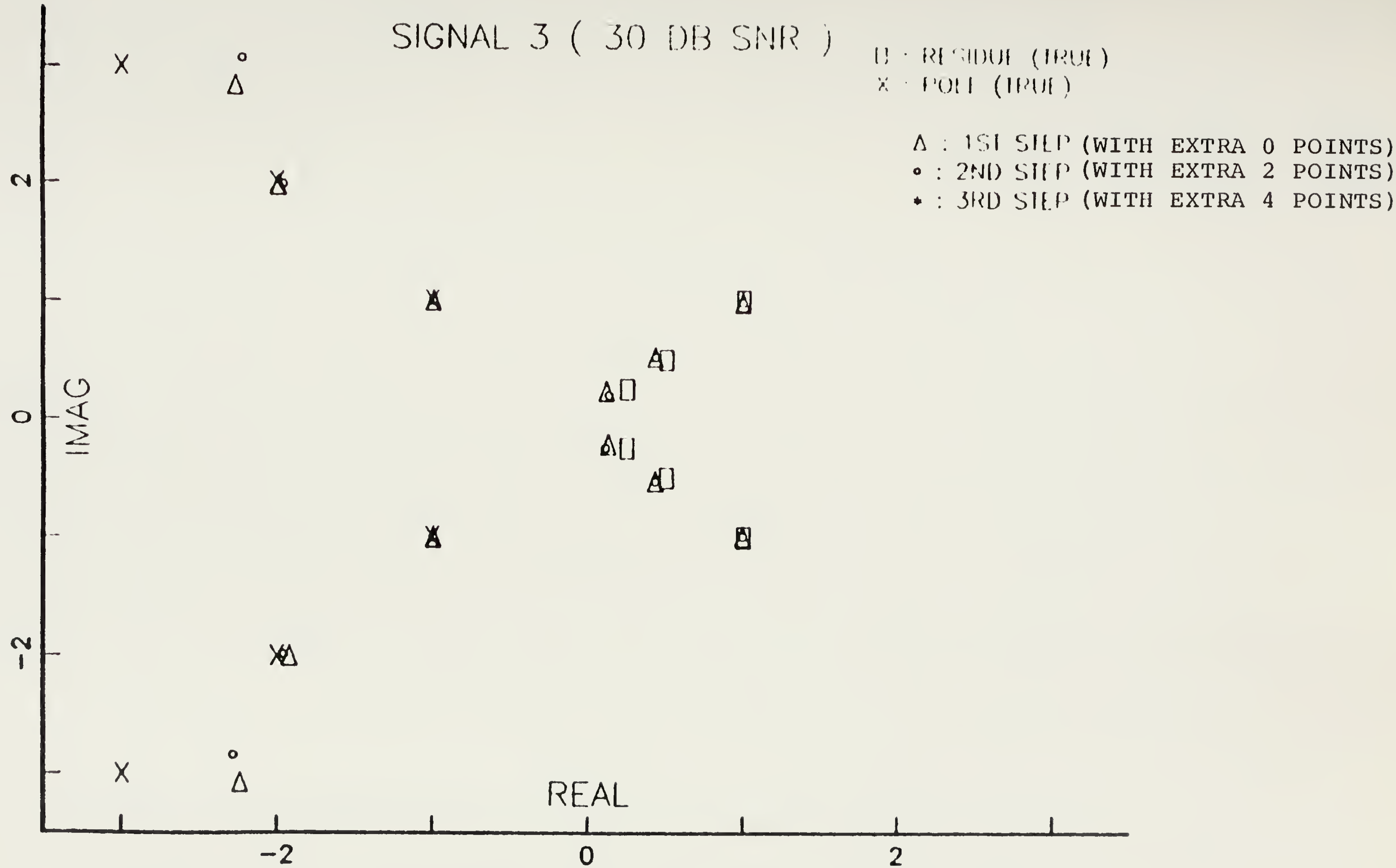

Figure 5.10. Pole and Residue Plot for Signal 3 of 30 dB SNR.

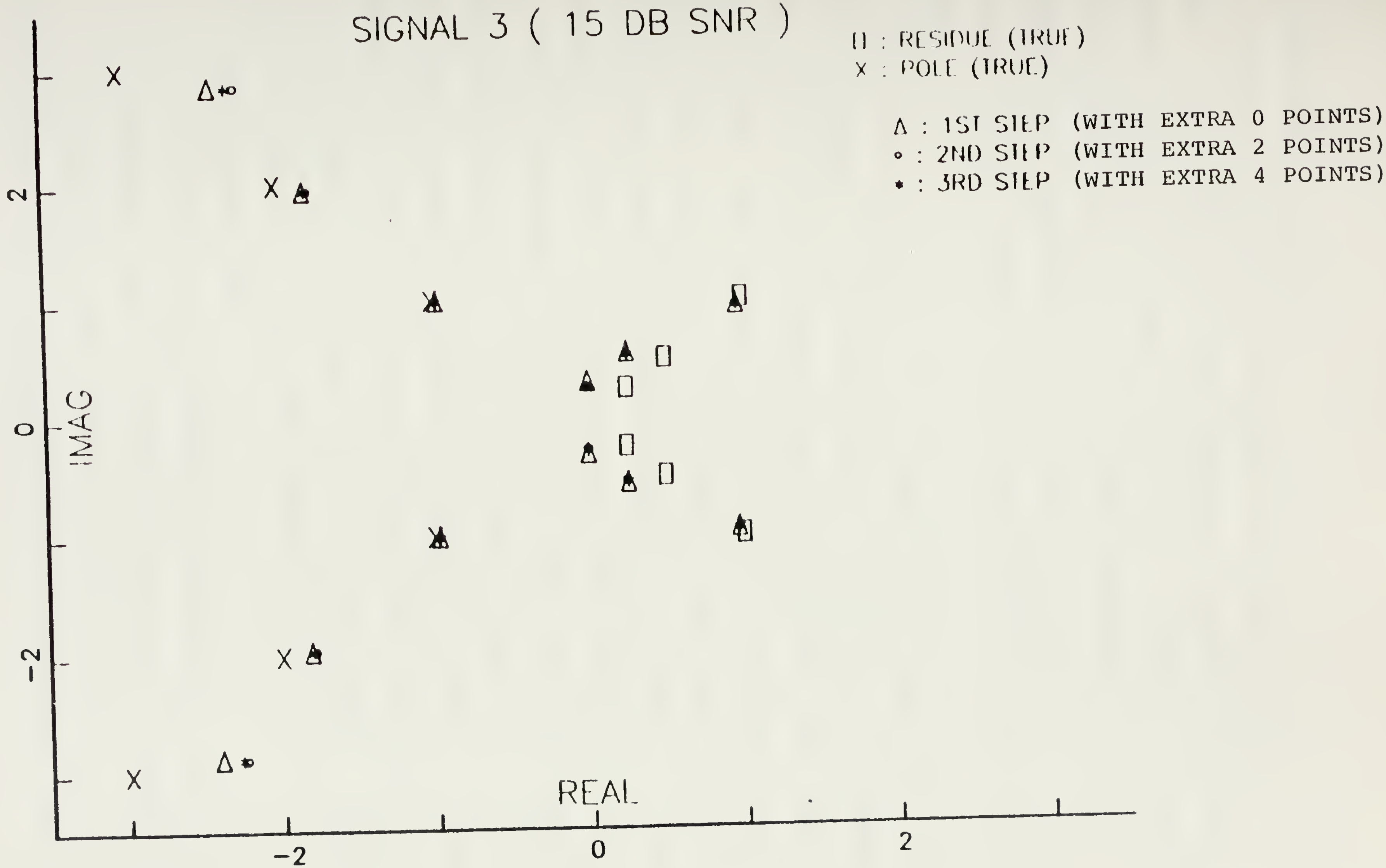

Figure 5.11. Pole and Residue Plot for Signal 3 of 15 dB SNR.

C. PERFORMANCE EVALUATION

Unlike the traditional methods, this method revealed itself as one that can be applicable to a general scattering problem, including the early time data. A potential superiority of this method over the traditional methods is found in its capability of handling the early time signal that has relatively large energy content and less sensitivity to the fixed noise level.

Regarding the capability of handling the early time signal, the performance of this method seems to be very high compared with the traditional methods which do not model that signal generically. But, two points need to be considered when we ensure this high degree of performance. One important point is that we be able to estimate properly the number of poles to optimally ask for. This will depend upon the SNR of the data, increasing generally with increasing SNR. The actual physical scattering data has an infinite number of poles but, because the energy content of natural modes tends to drop off rapidly beyond the bandwith of the excitation signal the higher order poles are ignored, blanding with the noise present. Another point is that the initial data window has to contain only the late time signal data.

As the results of the tables in the previous pages indicate, converging characteristics of the value of parameters is not blocked by a high level of noise pollution. It has been observed that the computed value of poles are likely to

converge closely to the true values that are listed in Tables I through III either by choosing the number of poles as small or by increasing the size of the data window. Even though it is hard to prove this optimality of this method (due to the non-linearity of the scattering transient response signal model), handling of the early time signal seems to be cast into a simpler problem space and that optimality may be explained by observing the parameters at every processing step. Namely, the parameter estimation of this method is less influenced by the unreliable points than the more accurate ones. Figure 5.1 and 5.2 and the results in the previous section explain the reason why the above assertion is valid. We see that the estimated value of parameters through optimization is not going to worse as we include the early time data points (these data points can be regarded as the unreliable data when we extrapolate the sum of exponentials of the late time region into those of the early time region) into the data window what we see. But, as we are adding more data points of the late time region, improved values of parameters are obtained through processing, as in the results and Fig. 5.2. Although this is true, we still see that the plot of E(k) in Fig. 5.2 does not provide sufficient data to confirm that an improvement of accuracy depends only on the number of late time data points, because those data points have a lower level of SNR locally. Now, let us get back to Tables IV, VII, and X. For the signal inputs whose SNRs are assumed to be

infinite (noise free), we have a set of parameters whose values are not converging to the true values as we add an increased number of unknowns. These results ensure us that the increased computer round-off errors obtained by more data processing (more parameters to extract) will produce divergence. Even initially with low SNR, the increased signal energy in the additive early-time data may improve the results until, by adding enough extra data points and parameters, round-off error will eventually catch up with the most optimized one and beyond a specific point the results will begin to diverge. So that, in a general sense, we can say that the improvement in accuracy may be obtained either by the number of data points or by making use of the data points of high SNRs. If we consider a signal having high damping coefficients we can not use the signal data what are likely to be hidden by noise. The technique used here in processing the data proceeds by sliding the first data point along the time-axis towards the origin until the data window expands its range up to time-origin by adding the unknown parameters E(k)'s which is the same as the number of data points in the early time region.

Figures 5.3 through 5.11 shows how rapidly the parameters we want to derive converge to the true values, within a convergence bounds. Here, all the plots were reconstructed from the results.

VI. SUMMARY

A modified least-squares minimization method has been developed and tested against synthetically generated signal data using a non-linear parameter optimization algorithm. The effect of varying the number of data points either in the region of early time or in the late time under various typical noise environments has been studied, defining the criteria for using this algorithm for the extraction of poles and residues under relatively heavy background noise, as well as for the time varying residue signals modeled by Morgan [Ref. 4].

In an attempt to use this method for signals with high damping coefficients, 3 synthetically generated signals were used as the inputs (Appendix C). Extraction of poles from the direct synthetic signal waveform was done through non-linear numerical evaluations, not by the Prony-based evaluations which were used in the traditional method. The parameter estimating algorithm for this non-linear signal model had worked successfully, providing users with optimized parameters that were converging to the true values, within specified convergence bounds. It was found that although we have the early time signal data having very high SNR, those data can not contribute to improving accuracy by simply increasing the number of extra data points and extra unknown parameters.

In order to get more accurate values of poles, we have to have an increased base of late time signal data, which have relatively low SNR.

It has been shown that this method is "robust" even under heavy noise conditions. But three basic requirements have to be met for using this method as a general methodology; the optimal number of requested poles has to be known in advance to processing, reasonable initial estimation of parameters has to be made, and the transition time from early to late time signal models has to be known a priori.

APPENDIX A

SYNTHETIC DATA GENERATION PROGRAM

```
C
C     *****************************************************************
C     *** DGEN - MAIN PROGRAM : GENERATES SYNTHETIC TRANSIENT RESPONSE***
C     ***                       DATA OF 512 POINTS                  ***
C     *** FOR THESIS RESERACH   DEC 20,1983 (#590) CHONG, CHOONG YEON ***
C     *****************************************************************
C
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION AGDF(5001)
      DIMENSION R1(64),R2(64),P1(64),P2(64)
      DIMENSION X0(512),XNO(512)
      DATA IY/'Y'/,IZ/'N'/
      DATA             D2P31M/2147483647.D0/
      DATA             D2P31/2147483648.D0/
C
C ====================================================================
C CLEAR SCREEN
      CALL FRTCMS('CLRSCRN ')
C
C DATA GENERATION MODE SELECTION
      WRITE(6,61)
   61 FORMAT(1X,///,10X,'*** SYNTHETIC SIGNAL DATA GENERTION ***',
     *//,     10X,'INTERACTIVE PROGRAM EXECUTION BEGINS ..............',
     *//,     10X,'**************************************************',
     */,      10X,'*    THIS PROGRAM GENERATES  512 POINTS OF DIGITAL *',
     */,      10X,'*  SIGNAL (TRADITIONAL OR NEW MODEL) SYNTHETICALLY *',
     */,      10X,'*  EITHER IN  NOISE-FREE OR  NOISE POLLUTING MODE. *',
     */,      10X,'**************************************************',
     *//,     10X,'USER IS ASKED TO INPUT THE DATA GERERATING PARAMETERS
     * INTERACTIVELY',/,1X,'TYPE <G> TO GO IF YOU READY TO INPUT')
      READ(5,51) IANS
   51 FORMAT(A1)
C
C INITIALIZATION. MAXIMUM 64 POLES
      DO 10 I=1,64
         R1(I)=0
         R2(I)=0
         P1(I)=0
   10    P2(I)=0
C
C GET THE DATA IDENTIFIERS
      CALL HEADER(NT1,NT2,NW1,NW2,ND1,ND2,NF1,NF2)
C
C GET THE DATA PARAMETERS
      CALL PARA(N1,T2,S1)
C
C GET THE DATA PARAMETERS-POLES/RESIDUES INTERACTIVELY
      WRITE(6,311)
```

```
  311 FORMAT(1X,'DO YOU WANT TO GET TRUE POLE/ZEROS FROM THE PRE-DEFINED
     * DATA FILE? -<Y/N>')
      READ(5,51) IANS
      IF(IANS.NE.IY) GO TO 312
      CALL DATA(N1,R1,R2,P1,P2)
      GO TO 313
C
  312 CALL RPIN(N1,R1,R2,P1,P2)
C
C VERIFICATION POINT - DISPLAY OF POLE AND RESIDUE TABLE
  313 WRITE(6,62)
   62 FORMAT(1X,'DO YOU WANT TO HAVE THE RES/POLES IN A TABLE FORM? - <Y
     */N>')
      READ(5,51) IANS
      IF(IANS.NE.IY) GO TO 990
      CALL FRTCMS('CLRSCRN ')
      CALL DISP1(NT1,NT2,NW1,NW2,ND1,ND2,NF1,NF2,N1,R1,R2,P1,P2)
  990 CONTINUE
C
C NOISE-FREE SYNTHETIC DATA GENERATOR
      WRITE(6,63)
   63 FORMAT(1X,'DO YOU WANT TO GENERATE A NOISE-FREE SIGNAL? - <Y/N>')
      READ(5,51) IANS
      IF(IANS.NE.IY) GO TO 991
      CALL GNFD(N1,R1,R2,P1,P2,T2,S1,M9,X0)
  991 CONTINUE
C
      CALL HEADER(NT1,NT2,NW1,NW2,ND1,ND2,NF1,NF2)
C
C NOISE-FREE DATA SIGNAL DISPLAY
      WRITE(6,58)
   58 FORMAT(1X,'DO YOU WANT TO SEE THE NOISE FREE SIGNAL IN A TABLE FOR
     *M? - <Y/N>')
      READ(5,51) IANS
      IF(IANS.NE.IY) GO TO 992
      N2=0
      CALL DISP2(NT1,NT2,NW1,NW2,ND1,ND2,NF1,NF2,N2,T2,S1,X0)
  992 CONTINUE
      IFLAG=0
      GO TO 5111
C
C NOISE POLLUTED SYNTHETIC DATA GENERATOR
 6111 IFLAG=1
      WRITE(6,59)
   59 FORMAT(1X,'DO YOU WANT TO GENERATE A NOISE-POLLUTED SIGNAL DATA? -
     * <Y/N>')
      READ(5,51) IANS
      IF(IANS.EQ.IZ) GO TO 9999
```

```
C
C GENERATE THE GAUSSIAN DISTRIBUTION
      CALL GAUSS(AGDF,IWA,IGA)
      GO TO 444
C
C OPTIONS FOR POLLUTING MODE
  444 WRITE(6,64)
   64 FORMAT(1X,'CHOOSE ONE OF THE THREE MODES OF NOISE POLLUTION',
     *//,     1X,'1 :  IN TERMS OF SNR(AVERAGE) IN DECIBEL',
     *//,     1X,'2 :  IN TERMS OF SNR(PEAK) IN DECIBEL',//)
      READ(5, *) IR
C
C ADJUST NOISE LEVEL IN QUICK
      CALL TRAP(X0,Q,Q1)
C
      WRITE(6,641) Q
  641 FORMAT(1X,'AVERAGE SIGNAL POWER : ',F12.6 )
      IF(IR.EQ.2) GO TO 222
      IF(IR.GT.2) GO TO 444
C
      WRITE(6,66)
   66 FORMAT(1X,'ENTER  SNR(AVERAGE) IN DECIBEL')
      READ(5,*)W1
      W3=10.**(W1/10.)
      DEV=DSQRT(Q/W3)
      GOTO 4111
C
  222 WRITE(6,67)
   67 FORMAT(1X,'ENTER  SNR(PEAK) IN DECIBEL')
      READ(5,*) W2
      W3=10.**(W2/10.)
      DEV=DSQRT(Q1/W3)
C
 4111 CONTINUE
C
C SEED OF RANDOM NOISE GENERATOR WITH UNIFORM DISTRIBUTION
      DSEED=12457.D0
C
C DO NOISE AVERAGING AND ADJUST IF NECESSARY
      WRITE(6,68)
   68 FORMAT(1X,'ENTER THE NUMBER OF NOISE AVERAGINGS - RECOMMEND:100')
      READ(5,*) N2
      DO 20 I=1,N2
         DO 201 J=1,512
            IS1=0
C           RANDOM NUMBER GENERATION
            DSEED = DMOD(16807.D0*DSEED,D2P31M)
            RV = DSEED / D2P31
```

```
C           GET GCF INDEPENDENT VARIABLE
            RN=RV
            IF(RN.LE.0.5) GOTO 555
            RN=1-RN
            IS1=1
C
  555       CALL TRANF(AGCF,IWA,DEV,RN,X)
C
            IF(IS1.EQ.0) GOTO 666
            XS=-X
            GOTO 777
  666       XS=X
  777       XNO(J)=XNO(J)+XS
  201    CONTINUE
   20 CONTINUE
C
      CALL NTEST(IR,W1,W2,Q,Q1,XNO,NFLAG,N2)
C
C ADD NOISE
      DO 30  I=1,512
   30    XO(I)=XO(I)+XNO(I)
C
C CHANGE IF NECESSARY
      CALL HEADER(NT1,NT2,NW1,NW2,ND1,ND2,NF1,ND2)
C
C NOISE POLLUTED SIGNAL DISPLAY
      WRITE(6,69)
   69 FORMAT(1X,'DO YOU WANT TO HAVE THE NOISE POLLUTED DATA IN A TABLE
     * FORM? - <Y/N>')
      READ(5,51)IANS
      IF(IANS.EQ.IZ)  GOTO 9999
C
      CALL DISP2(NT1,NT2,NW1,NW2,ND1,ND2,NF1,NF2,N2,T2,S1,XO)
C
C SYNTHETIC DATA STORAGE
C
 5111 WRITE(6,71)
   71 FORMAT(1X,'ENTER THE FILE NUMBER OF A DATA FILE INTO WHERE YOU WAN
     *T TO STORE THE DATA',/,1X,'IF YOU DO NOT WANT TO, ENTER 13',/,
     */,1X,'1 : HIGH-DAMPING/NOISE-FREE/CONSTANT RESIDUE DATA FILE',
     */,1X,'2 :                            /TIME VARYING RESIDUE DATA FILE',
     */,1X,'3 :                  /30DB-SNR  /TIME VARYING RESIDUE DATA FILE',
     */,1X,'4 :                  /15DB-SNR  /TIME VARYING RESIDUE DATA FILE',
     */,1X,'5-6 : NO NOT USE',
     */,1X,'7 : LOW-DAMPING/NOISE-FREE/CONSTANT RESIDUE DATA FILE',
     */,1X,'8 :                            /TIME VARYING RESIDUE DATA FILE',
     */,1X,'9 :                  /30DB-SNR  /TIME VARYING RESIDUE DATA FILE',
     */,1X,'10:                  /15DB-SNR  /TIME VARYING RESIDUE DATA FILE')
```

```
C
      READ(5,*) IFILE
      IF(IFILE.EQ.13) GO TO 7
      WRITE(IFILE,8) NT1,NT2,NW1,NW2,ND1,ND2,NF1,NF2,N2,T2,S1
    8 FORMAT(2X,2A4,/,2X,2A4,/,2X,2A4,/,
     *       2X,2A4,/,2X,I4,/,2X,F14.8,/,2X,F14.8)
      DO 7 I=1,512,4
         WRITE(IFILE,9) XO(I),XO(I+1),XO(I+2),XO(I+3)
    9    FORMAT(2X,4(2X,F14.8))
    7 CONTINUE
      IF(IFLAG.EQ.0) GO TO 6111
  995 CONTINUE
C
  111 CONTINUE
 9999 CONTINUE
C
      CALL FRTCMS('CLRSCRN ')
C
      WRITE(6,72)
   72 FORMAT(1X,/////,10X,'***  PROCESSING COMPLETED  ***')
C
      STOP
      END
C
C
C     ================================================================
C     = SUBROUTINE HEADER - CAN BE USED WHENEVER USER NEEDS TO CHANGE =
C     ================================================================
C
      SUBROUTINE HEADER(NT1,NT2,NW1,NW2,ND1,ND2,NF1,NF2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DATA IY/'Y'/,IZ/'N'/
C
      CALL FRTCMS('CLRSCRN ')
C
   59 WRITE(6,60)
   60 FORMAT(1X,'DO YOU WANT TO INITIALIZE/CHANGE THE HEADER? - <Y/N>')
      READ(5,50) IANS
      IF(IANS.EQ.IZ) GO TO 10
      IF(IANS.NE.IY) GO TO 59
C
      WRITE(6,61)
   61 FORMAT(1X,'ENTER THE TARGET TYPE WITHIN 8 CHARS',
     */,1X,'EX : TGT-1')
      READ(5,51) NT1,NT2
      WRITE(6,62)
   62 FORMAT(1X,'ENTER THE WAVEFORM TYPE WITHIN 8 CHARS',
```

```
     */,1X,'EX : HDTN - HIGH DAMPING/TIME VARYING RESIDUE/NOISE ADDED',
     */,1X,'       LDCF - LOW DAMPING/CONSTANT RESIDUE/NOISE FREE')
      READ(5,51) NW1,NW2
      WRITE(6,63)
   63 FORMAT(1X,'ENTER THE DATE WITHIN 8 DIGITS',
     */,1X,'EX : DEC. 20')
      READ(5,51) ND1,ND2
      WRITE(6,64)
   64 FORMAT(1X,'ENTER THE NAME OF FILE WITHIN 8 CHARS',
     */,1X,'EX : FILE001')
      READ(5,51) NF1,NF2
   50 FORMAT(A1)
   51 FORMAT(2A4)
C
   10 CONTINUE
C
      RETURN
      END
C
C
C     ==================================================================
C     = SUBROUTINE PARA - INITIALIZE THE NUMBER OF POLES, SIZE OF TIME =
C     =                   WINDOW AND VERTICAL SCALE IN GRAPHIC         =
C     ==================================================================
C
      SUBROUTINE PARA(N1,T2,S1)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DATA IB/'B'/,IG/'G'/
C
    1 CALL FRTCMS('CLRSCRN ')
C
      WRITE(6,60)
   60 FORMAT(1X,'ENTER  THE  NUMBER  OF  POLES : MAX 64')
      READ(5,*) N1
      WRITE(6,61)
   61 FORMAT(1X,'ENTER THE SIZE OF TIME WINDOW : 20 NANO SECOND RIGID')
      READ(5,*) T2
      WRITE(6,62)
   62 FORMAT(1X,'ENTER A VERTICAL SCALE FACTOR : 1 INITIALLY')
      READ(5,*) S1
C
    2 WRITE(6,69)
   69 FORMAT(1X,'TYPE <B> TO CHANGE THE ABOVE INITIALIZATIONS',
     */,       1X,'OTHERWISE, TYPE <G> TO GO ON')
      READ(5,51) IANS
   51 FORMAT(A1)
      IF(IANS.EQ.IB) GO TO 1
```

```
      IF(IANS.NE.IG) GC TC 2
C
      RETURN
      END
C
C
C    ==================================================================
C    = SUBROUTINE RPIN - INPUT THE DATA GENERATING PARAMETER          =
C    ==================================================================
C
      SUBROUTINE RPIN(ISN1,SR1,SR2,SP1,SP2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSICN SR1(64),SR2(64),SP1(64),SP2(64)
C
      CALL FRTCMS('CLRSCRN ')
C
      WRITE(6,60)
   60 FORMAT(1X,'YOU ARE ASKED TO INPUT INTERACTIVELY THE VALUE OF PARAM
     *ETERS')
      DO 10 I=1,ISN1
         WRITE(6,61) I, I, I, I
   61 FORMAT(1X,'R-RE(',I2,')=',2X,'R-IM(',I2,')=',
     *       2X,'P-RE(',I2,')=',2X,'P-IM(',I2,')=')
   1C    REAC(5,*) SR1(I),SR2(I),SP1(I),SP2(I)
      RETURN
      ENC
C
C
C    ==================================================================
C    = SUBROUTINE DISP1 - TABULATES THE STATUS OF PARAMETERS          =
C    ==================================================================
C
      SUBROUTINE CISP1(NST1,NST2,NSW1,NSW2,NSD1,NSD2,NSF1,NSF2,
     *                 NSN1,SR1,SR2,SP1,SP2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSICN SR1(64),SR2(64),SP1(64),SP2(64)
C
    1 CALL FRTCMS('CLRSCRN ')
C
      WRITE(6,311)
  311 FORMAT(1X,'ENTER THE FILE LOCAITON INTO WHERE THE POLE AND RESIDUE
     * DATA ARE STORED',
     */,1X,'11 : RESERVED FOR HIGH DAMPING DATA',
     */,1X,'12 : RESERVEC FOR LOW  DAMPING DATA',/)
C
      READ(5,*) IFILE
```

```
C
      WRITE(IFILE,59) NST1,NST2,NSW1,NSW2,NSD1,NSD2,NSF1,NSF2,NSN1
      WRITE(6,59) NST1,NST2,NSW1,NSW2,NSD1,NSD2,NSF1,NSF2,NSN1
   59 FORMAT(2X,'TARGET    TYPE:',2A4,
     */,      2X,'WAVEFORM TYPE:',2A4,
     */,      2X,'CONTACT   DATE:',2A4,
     */,      2X,'FILE      NUMB:',2A4,
     */,      2X,'NUMB. OF POLE:',I4,/)
C
      WRITE(IFILE,60)
C
      WRITE(6,60)
   60 FORMAT(2X,15X,'TABLE OF RESIDUES AND POLES',
     */,      2X,15X,'===========================',/,
     */,2X,'PAIR #',4X,' RES-REAL',4X,' RES-IMAG',
     *                 4X,'POLE-REAL',4X,'POLE-IMAG')
      DO 10 I=1,NSN1
C
         WRITE(IFILE,61) I,SR1(I),SR2(I),SP1(I),SP2(I)
C
         WRITE(6,61) I,SR1(I),SR2(I),SP1(I),SP2(I)
   61    FORMAT(2X,I4,3X,4(F12.8,1X))
   10 CONTINUE
      RETURN
      END
C
C
C    ===============================================================
C    = SUBROUTINE GNFD - GERERATES NOISE FREE SIGNAL               =
C    =                   6 OPTIONS FOR RESIDUE FUNCTION SELECTION  =
C    ===============================================================
C
      SUBROUTINE GNFD(NN1,SR1,SR2,SP1,SP2,ST2,SS1,M9,SX0)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION SR1(64),SR2(64),SP1(64),SP2(64),SX0(512),POWER(10)
      DATA IG/'G'/,IB/'B'/
C
      PI=3.141592654
C
C
    1 CALL FRTCMS('CLRSCRN ')
C
  111 WRITE(6,60)
   60 FORMAT(1X,'CHOOSE ONE OF THE FOLLOWING ENTIRE FUNCTION',
     *//,     1X,'OPTION 4 IS RECOMMENDED RIGOROUSLY',
     *//,     2X,5X,'1 : DC ',
     */,      2X,5X,'2 : POSITIVE LINEAR',
```

```
     */,        2X,5X,'3 : NEGATIVE LINEAR',
     */,        2X,5X,'4 : TRAPEZOIDAL',
     */,        2X,5X,'5 : CONSTANT',
     */,        2X,5X,'6 : TRIANGULAR')
      SS1=1.
      READ(5,*) M9
      IF(M9.GT.6) GOTO 111
      IF(M9.EQ.5) GOTO 999
C
      WRITE(6,61)
   61 FORMAT(1X,'SPECIFY THE POINT AT WHICH E(K) IS TO BE SET TO ZERO',
     */,        1X,'RECOMMENDED 10 : WHEN E(K) : A TRAPEZOIDAL')
      READ(5,*) M3
      GO TO 888
C
  999 M3=1
  888 TO=ST2/511.
C
      T9=0.
      DO 10 I=1,M3
         X9=0.
         X8=0.
         DO 20 J=1,NN1
            X8=SR1(J)*DCOS(2*PI*SP2(J)*T9)-SR2(J)*DSIN(2*PI*SP2(J)*T9)
   20       X9=X9+X8*DEXP(SP1(J)*T9)
         SXO(I)=X9
         T9=T9+TO
         POWER(I)=X8*2
   10 CONTINUE
      TP=0
      DO 40 I=1,M3
   40    TP=TP+POWER(I)**2
C
    4 WRITE(6,73)
   73 FORMAT(1X,'TYPE <B> IF YOU WANT TO CHOOSE ANOTHER OPTION',
     */,        1X,'OTHERWISE TYPE <G> TO GO')
      READ(5,51) IANS
      IF(IANS.EQ.IB) GO TO 1
      IF(IANS.NE.IG) GO TO 4
C
      CALL FRTCMS('CLRSCRN ')
C
      IF(M9.EQ.5) GO TO 99
      IF(M9.EQ.1) GO TO 88
      IF(M9.EQ.2) GO TO 77
      IF(M9.EQ.3) GO TO 66
      IF(M9.EQ.4) GO TO 55
C
```

```
C TRIANGULAR FUNCTION OF E(K)
      XTR6=DSQRT(TP/6.)
      SXO(2)=SXO(2) + XTR6
      SXO(3)=SXO(3) + 2*XTR6
      SXO(4)=SXO(4) + XTR6
      GO TO 99
C
C TRAPEZOIDAL FUNCTION OF E(K)
   55 XTR4=DSQRT(TP/60.)
      SXO(2)=SXO(2) + XTR4
      SXO(3)=SXO(3) + 2*XTR4
      SXO(4)=SXO(4) + 3*XTR4
      SXO(5)=SXO(5) + 4*XTR4
      SXO(6)=SXO(6) + 4*XTR4
      SXO(7)=SXO(7) + 3*XTR4
      SXO(8)=SXO(8) + 2*XTR4
      SXO(9)=SXO(9) + XTR4
      GO TO 99
C
C NEGITIVE LINEAR FUNCTION OF E(K)
   66 XTR3=DSQRT(TP/30.)
      SXO(1)=SXO(1) + 4*XTR3
      SXO(2)=SXO(2) + 3*XTR3
      SXO(3)=SXO(3) + 2*XTR3
      SXO(4)=SXO(4) + 1*XTR3
      GO TO 99
C
   77 XTR2=DSQRT(TP/30.)
      SXO(2)=SXO(2) + XTR2
      SXO(3)=SXO(3) + 2*XTR2
      SXO(4)=SXO(4) + 3*XTR2
      SXO(5)=SXO(5) + 4*XTR2
      GO TO 99
C DC FUNCTION OF E(K)
   88 XTR1=DSQRT(TP/5.)
      DO 317I=1,M3
  317    SXO(I)=SXO(I) + XTR1

   99 WRITE(6,63)
   63 FORMAT(1X,'SAMPLING UP TO M3 POINT COMPLETED...')
C
      MP=M3+1
      DO 30 I=MP,512
         X9=C.
         X8=C.
         DO 31 J=1,NN1
            X8=SR1(J)*DCOS(2*PI*SP2(J)*T9)-SR2(J)*DSIN(2*PI*SP2(J)*T9)
   31       X9=X9+X8*DEXP(SP1(J)*T9)
```

```
      SXO(I)=X9
      T9=T9+TO
   30 CONTINUE
C
    2 WRITE(6,64)
   64 FORMAT(1X,'SAMPLING UP TO 512 POINT COMPLETED...',
     */,       1X,'TYPE <G> TO GO ')
      READ(5,51) IANS
   51 FORMAT(A1)
      IF(IANS.EQ.IB) GO TO 1
      IF(IANS.NE.IG) GO TO 2
C
      RETURN
      END
C
C
C  ==================================================================
C  = SUBROUTINE TRAP - CALCULATES PEAK AND AVERAGE SIGNAL POWER     =
C  ==================================================================
C
      SUBROUTINE TRAP(SXO,AVG,PEAK)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION SXO(512)
C
      PSUM=0.
      FIRST=SXO(1)
      PSUM=SXO(1)**2
      DO 10 I=2,512
         XTM2=SXO(I)**2
         XTM1=SXO(I-1)**2
         PSUM=PSUM+XTM2
   10    IF(XTM2.GT.XTM1) FIRST=SXO(I)
C
      AVG=PSUM/511.
      PEAK=FIRST**2
C
      RETURN
      END
C
C
C  ==================================================================
C  = SUBROUTINE GAUSS - EVALUATES 5000 POINTS GAUSSIAN DISTRIBUTION =
C  =                    IN ROUGH MANNER.                            =
C  ==================================================================
C
      SUBROUTINE GAUSS(AGDF,IWA,IGA)
C
```

```
      IMPLICIT REAL*8(A-H,O-Z)
      DATA IY/'Y'/,IZ/'N'/
      INTEGER IWA,IGA
      DIMENSION AGDF(5001)
C
      CALL FRTCMS('CLRSCRN ')
C
      PI=3.141592654
      WRITE(6,60)
   60 FORMAT(1X,'ENTER THE NUMBER OF SAMPLING POINTS OF GAUSSIAN PDF'
     *,/,      1X,'RECOMMENDED : 5001,(4376 OR 3751:OPTIONAL)')
      READ(5,*) IWA
C
      DO 1 I=1,5001
    1    AGDF(I)=0.
C
      IGA=IWA-1
      RDEL=5./DFLOAT(IGA)
      RRES=1./DSQRT(2.*PI)
      AGDF(1)=RRES*EXP(0.)
      RBAS=RRES
C
      DO 10  I=2,IWA
         REXP=RBAS**2./2.
         AGDF(I)=RRES*DEXP(-REXP)
   10    RBAS=RDEL*I
C
      AFIR=AGDF(1)
      AGDF(1)=0.5
      DO 20  J=2,IWA
         A=AGDF(J)
         AGDF(J)=AGDF(J-1)+0.5*(AFIR+A)*5./DFLOAT(IGA)
   20    AFIR=A
C
  111 WRITE(6,62) IGA
   62 FORMAT(1X,I5,' POINTS GAUSSIAN DF WERE EVALLATED',/,
     *           1X,'DO YCU WANT TO SEE F(X) TABLE? - <Y/N>')
      READ(5,51) IANS
   51 FORMAT(A1)
      IF(IANS.EQ.IY) GOTC 112
      IF(IANS.EQ.IZ) GOTO 113
      GO TO 111
C
  112    CALL FRTCMS('CLRSCRN ')
C
         WRITE(6,621)
  621    FORMAT(2X,'   X        F(X)        X       F(X)        X
     *F(X)',/,2X,'======  ==========   ======  ==========   ======   ===
```

```
     *=======')
      K=1
C
  114 X1=(K-1)*RDEL
      X2=K*RDEL
      X3=(K+1)*RDEL
      WRITE(6,622) X1,AGDF(K),X2,AGDF(K+1),X3,AGDF(K+2)
      K=K+3
      IF(K.LE.(IGA-2)) GO TO 114
  622 FORMAT(1X,F7.3,2X,D10.5,2X,F7.3,2X,D10.5,2X,F7.3,2X,D10.5)
C
  113 CONTINUE
C
  222 WRITE(6,63)
   63 FORMAT(1X,'DO YOU WANT TO STORE THE F(X) TABLE INTO DATA FILE? - <
     *Y/N>')
      READ(5,51) IANS
      IF(IANS.EQ.IY) GOTO 223
      IF(IANS.EQ.IZ) GOTO 224
      GO TO 222
C
  223     WRITE(8,81) IWA
   81     FORMAT(2X,D14.9)
C
          DO 40 M=1,5000
   40        WRITE(8,81) AGDF(M)
C
          WRITE(6,64)
   64     FORMAT(1X,'DATA WERE STORED...')
C
  224 WRITE(6,65)
   65 FORMAT(1X,'TYPE <G> TO GO')
      READ(5,51) IANS
C
      CALL FRTCMS('CLRSCRN ')
C
      RETURN
      END
C
C
C    ================================================================
C    = SUBROUTINE TRANF - TRANSFORM THE UNIFORM NOISE INTO PSUDO -  =
C    =                    GAUSSIAN NOISE.                           =
C    ================================================================
C
      SUBROUTINE TRANF(AGDF,IWA,SDEV,SRN,SX)
C
      IMPLICIT REAL*4(A-H,O-Z)
```

```
      DIMENSION AGDF(5001)                                              DGE06250
C                                                                       DGE06260
      IGA=IWA-1                                                         DGE06270
      IF(AGDF(1).EQ.SRN) GO TO 99                                       DGE06280
      IF(AGDF(IWA).LE.SRN) GO TO 98                                     DGE06290
C                                                                       DGE06300
      L=0                                                               DGE06310
      TEMP=IGA*.2                                                       DGE06320
      L2=INT(TEMP)                                                      DGE06330
      L1=L2                                                             DGE06340
   12 IF(AGDF(L1).EQ.SRN) GOTO 88                                       DGE06350
      IF(AGDF(L1).GT.SRN) GOTO 22                                       DGE06360
      L=L1                                                              DGE06370
      L1=L1+L2                                                          DGE06380
      GO TO 12                                                          DGE06390
C                                                                       DGE06400
   22 TEMP=L2*.2                                                        DGE06410
      L3=INT(TEMP)                                                      DGE06420
      L1=L+L3                                                           DGE06430
   23 IF(AGDF(L1).EQ.SRN) GO TO 88                                      DGE06440
      IF(AGDF(L1).GT.SRN) GO TO 33                                      DGE06450
      L=L1                                                              DGE06460
      L1=L1+L3                                                          DGE06470
      GO TO 23                                                          DGE06480
C                                                                       DGE06490
   33 TEMP=L3*.2                                                        DGE06500
      L4=INT(TEMP)                                                      DGE06510
      L1=L+L4                                                           DGE06520
   34 IF(AGDF(L1).EQ.SRN) GO TO 88                                      DGE06530
      IF(AGDF(L1).GT.SRN) GO TO 44                                      DGE06540
      L=L1                                                              DGE06550
      L1=L1+L4                                                          DGE06560
      GO TO 34                                                          DGE06570
C                                                                       DGE06580
   44 TEMP=L4*.2                                                        DGE06590
      L5=INT(TEMP)                                                      DGE06600
      L1=L+L5                                                           DGE06610
   45 IF(AGDF(L1).EQ.SRN) GOTO 88                                       DGE06620
      IF(AGDF(L1).GT.SRN) GOTO 55                                       DGE06630
      L=L1                                                              DGE06640
      L1=L1+L5                                                          DGE06650
      GO TO 45                                                          DGE06660
C                                                                       DGE06670
   55 L1=L+1                                                            DGE06680
   56 IF(AGDF(L1).GE.SRN) GOTO 88                                       DGE06690
      L1=L1+1                                                           DGE06700
      GO TO 56                                                          DGE06710
C                                                                       DGE06720
```

```
99 SX=0.
   GO TO 77
C
98 SX=SDEV*5.
   GO TO 77
C
88 SX=L1*5./DFLOAT(IGA)*SDEV
C
77 RETURN
   END
C
C
C  ================================================================
C  = SUBROUTINE DISP2 - DISPLAYS THE DATA SIGNAL IN A TABLE FORM  =
C  ================================================================
C
   SUBROUTINE DISP2(NT1,NT2,NW1,NW2,ND1,ND2,NF1,NF2,N2,T2,S1,XO)
C
   IMPLICIT REAL*8(A-H,O-Z)
   DIMENSION XO(512)
C
   CALL FRTCMS('CLRSCRN ')
C
   WRITE(6,60) NT1,NT2,NW1,NW2,ND1,ND2,NF1,NF2,N2,T2,S1
60 FORMAT(2X,'TARGET    TYPE:',2A4,
  */,      2X,'WAVEFORM  TYPE:',2A4,
  */,      2X,'CONTACT   DATE:',2A4,
  */,      2X,'FILE      NUMB:',2A4,
  */,      2X,'AVERAGINGS(#):',I4,
  */,      2X,'TIME    WINDOW:',F14.8,
  */,      2X,'VERTI.   SCALE:',F14.8,/)
   WRITE(6,61)
61 FORMAT(2X, 2(5X,'T',4X,5X,5X,'X(T)',4X,5X)
  */,      2X, 2(2X,8('='),  5X,2X,12('='),5X))
C
   XDEL=T2/511.
   DO 10 I=1,512,2
      XINT1=XDEL*(I-1)
      XINT2=XDEL*I
      WRITE(6,62) XINT1,XO(I),XINT2,XO(I+1)
62    FORMAT(2X,F10.6,5X,F14.8,5X,F10.6,5X,F14.8)
10 CONTINUE
C
   RETURN
   END
C
C
C  ================================================================
```

```
C     = SUBROUTINE NTEST - TEST AND ADJUST THE NOISE LEVEL ROUGHFLY     =
C     ====================================================================
C
      SUBROUTINE NTEST(IR,W1,W2,Q,Q1,XNO,NFLAG,N2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XNO(512)
      DATA IY/'Y'/,IZ/'N'/
C
  234 NFLAG=0
      IF(IR.EQ.2) GO TO 2
      POWER=XNO(1)**2
C
      DO 10 I=2,512
   10    POWER = POWER + XNO(I)**2
      APOWER = POWER/511.
      RATIO=Q/APOWER
      IF(RATIO.GT.10E+6) RATIO =10E+6
      SNRDB=10*DLOG10(RATIO)
C
    1 WRITE(6,612) N2,SNRDB,W1
  612 FORMAT(1X,'RESULTED SNR(DB) FOR ',I4,'  AVERAGINGS :',F10.6,
     */,1X,'DO YOU WANT TO READJUST THE NOISE LEVEL TO ',F10.6,' DB AGAI
     *N <Y/N>')
      READ(5,51) IANS
   51 FORMAT(A1)
      IF(IANS.EQ.IZ) GO TO 19
      IF(IANS.NE.IY) GO TO 1
      DIF=SNRDB-W1
      IF(DIF.GE.0.) GO TO 4
C
C
      COEFF=-1./10**(-DIF/20.)
      GO TO 5
    4 COEFF= 10**(DIF/20.)
    5 CONTINUE
      AMAG=DSQRT(APOWER)
      BIAS=AMAG*COEFF
      DO 30 I=1,512
         IF(XNO(I).GE.0.0) XNO(I)=XNO(I)+BIAS
   30    XNO(I)=XNO(I)-BIAS
      GO TO 3
C
    2 PMAX=XNO(1)**2
      DO 20 I=2,512
         TMAX=XNO(I)**2
   20    IF(TMAX.GT.PMAX) PMAX=TMAX
C
```

```
      RATIO=C1/PMAX
      SNRDB=10*DLOG10(RATIO)
    6 WRITE(6,612) SNRDB,N2,W2
      READ(5,51) IANS
      IF(IANS.EQ.IZ) GO TO 3
      IF(IANS.NE.IY) GO TO 6
      DIF=SNRDB-W2
C
      IF(DIF.GE.0.) GO TO 7
      COEFF=-1./10**(-DIF/20.)
      GO TO 8
C
    7 COEFF=10**(DIF/20.)
      AMAG=DSQRT(PMAX)
      BIAS=AMAG*COEFF
C
    8 DO 40 I=1,512
         IF(XNO(I).GE.0.0) XNO(I)=XNO(I)+BIAS
   40    XNO(I)=XNO(I)-BIAS
C
    3 CONTINUE
      IF(NFLAG.EQ.0) GO TO 234
   19 CONTINUE
C
      RETURN
      END
C
C
C    ==================================================================
C    = SUBROUTINE CATA - PROVIDE A PRE-DEFINED SET OF POLES AND RESDUE =
C    ==================================================================
C
      SUBROUTINE CATA(N1,R1,R2,P1,P2)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION R1(64),R2(64),P1(64),P2(64)
C
C PREDEFINED PARAMETERS
      R1(1)=  1.0
      R2(1)=  1.0
      P1(1)= -1.0
      P2(1)=  1.0
      R1(2)=  1.0
      R2(2)= -1.0
      P1(2)= -1.0
      P2(2)= -1.0
C
      R1(3)=  0.5
```

```
      R2(3)=  0.5                                                       DGE08170
      P1(3)= -2.0                                                       DGE08180
      P2(3)=  2.0                                                       DGE08190
      R1(4)=  0.5                                                       DGE08200
      R2(4)= -0.5                                                       DGE08210
      P1(4)= -2.0                                                       DGE08220
      P2(4)= -2.0                                                       DGE08230
C                                                                       DGE08240
      R1(5)=  0.25                                                      DGE08250
      R2(5)=  0.25                                                      DGE08260
      P1(5)= -3.0                                                       DGE08270
      P2(5)=  3.0                                                       DGE08280
      R1(6)=  0.25                                                      DGE08290
      R2(6)= -0.25                                                      DGE08300
      P1(6)= -3.0                                                       DGE08310
      P2(6)= -3.0                                                       DGE08320
      RETURN                                                            DGE08330
      END                                                               DGE08340
```

APPENDIX B

NON-LINEAR PARAMETER OPTIMIZATION PROGRAM

```
C
C ****************************************************************************
C ***    PROC - MAINPROGRAM   : EXTRACT POLES AND RESIDUES OF A NEW        ***
C ***                           SIGNAL MODEL.                              ***
C ***    FOR THESIS RESEARCH, DEC 20,1983  (#590) CHONG, CHOONG YEON       ***
C ****************************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      EXTERNAL    FUNC
      DIMENSION   PARM(4),X(40),F(110),XJAC(110,40),XJTJ(1000),
     *            WORK(1000),Y(110),V(110),XO(512)
      COMMON      /ZSQ/Y,V
      DATA        IY/'Y'/,IZ/'N'/,IB/'B'/
C
C ==================================================================
C
C CLEAR SCREEN
      CALL FRTCMS('CLRSCRN ')
C
C HEADER
      WRITE(6,61)
   61 FORMAT(1X,//,10X,'*** NON-LINEAR PARAMETER OPTIMIZATION ***',
     */,       10X,'INTERACTIVE PROGRAM EXECUTION BEGINS ......',
     *//,      10X,'*****************************************',
     */,       10X,'** THIS PROGRAM CAN HANDEL MAXIMUM  19    **',
     */,       10X,'** PARAMETERS WITH 38 POINTS OF DATA SET **',
     */,       10X,'*****************************************',
     *//,10X,'TYPE <G> TO GO TO NEXT PAGE')
      READ(5,51) IANS
   51 FORMAT(A1)
C
      CALL FRTCMS('CLRSCRN ')
C
      WRITE(6,161)
  161 FORMAT(1X,
     *//,      10X,'ENTER THE FILE NUMBER OF A SIGNAL DATA FILE YOU WANT
     *TO READ',
     */,10X,'1 : HIGH-DAMPING/NOISE-FREE/CONSTANT-RESIDUE DATA FILE',
     */,10X,'2 :                          /TIME VARING RESIDUE DATA FILE',
     */,10X,'3 :              /30DB-SNR  /TIME VARING RESIDUE DATA FILE',
     */,10X,'4 :              /15DB-SNR  /TIME VARING RESIDUE DATA FILE',
     */,10X,'5 - 7 : DO NOT USE',
     */,10X,'8 : LOW-DAMPING/NOISE-FREE/CONSTANT-RESIDUE DATA FILE',
     */,10X,'9 :                          /TIME-VARING RESIDUE DATA FILE',
     */,10X,'10:               /30DB-SNR  /TIME-VARING RESIDUE DATA FILE',
     */,10X,'11:               /15DB-SNR  /TIME-VARING RESIDUE DATA FILE')
C
      READ(5,*) IFILE
```

```
C
C READ DATA FILE
      READ(IFILE,8) NT1,NT2,NW1,NW2,NC1,NC2,NF1,NF2,N2,T2,S1
      DO 10 I=1,512,4
  10     READ(IFILE,9) XO(I),XO(I+1),XO(I+2),XO(I+3)
   8 FORMAT( 2X,2A4,/,2X,2A4,/,2X,2A4,/,2X,2A4,/, 2X,I4,/,2X,F14.8,
    */,2X,F14.8)
   9 FORMAT(2X,4(2X,F14.8))
C
C INFORMATION
      WRITE(6,62)
  62 FORMAT(1X,'SIGNAL DATA FILE WERE READ...',
    */,      1X,'TYPE <G> TO GO')
      READ(5,51) IANS
C
C PARAMETER DEFAULTED (USER DEFINALBE)
      MAXFN=500
      IOPT=1
      NSIG=3
      EPS=0.0
      DELTA=0.0
C
C GET THE USER-DEFINED PARAMETERS
C M3 - INSTANT AT WHICH THE CONSTANT RESIDUE FUNCTION STARTS
C N1 - NUMBER OF POLES
C NE1 - INSTANT AT WHICH THE FIRST DATA ELEMENTS IN A DATA WINDOW
C
      CALL PARAS(M3,N1,NE1)
C
C VARIABLE & DIMENSION PARAMETER
C
 912 CALL FRTCMS('CLRSCRN ')
C
C INFORMATION OF STORAGE FOR DATA WRITING
      WRITE(6,311)
 311 FORMAT(1X,'ENTER THE FILE NUMBER OF THE DATA FILE TO WHERE THE DAT
    *A ARE STORED',/,
    */,1X,'20-29:RESERVED FOR H/LOW DAMPING NO NOISE SIGNAL PROCESSING'
    */,1X,'30-39:                           30DB SNR SIGNAL PROCESSING'
    */,1X,'40-49:                           15DB SNR SIGNAL PROCESSING'
    */,1X,'50-59:                           NO NOISE SIGNAL PROCESSING'
    */,1X,'60-69:                           30DB SNR SIGNAL PROCESSING'
    */,1X,'70-79:                           15DB SNR SIGNAL PROCESSING'
    */,1X,'       *0-WITH  0 EXTRA PARAMETER  E(K):2N  DATA POINTS',
    */,1X,'       *1-WITH  2 EXTRA PARAMETERS E(K):2N+2',
    */,1X,'       *2-WITH  4 EXTRA PARAMETERS E(K):2N+4',
    */,1X,'       *3-WITH  6 EXTRA PARAMETERS E(K):2N+6',
    */,1X,'       *4-WITH  8 EXTRA PARAMETERS E(K):2N+8',
```

```
     */,1X,'        *5-WITH 10 EXTRA PARAMETERS E(K):FULL DATA SETS',
     */,1X,'        *9-WITHOUT E(K) WITH THE ALL DATA IN A WINDOW')
C
      READ(5,*) IFILE
C
      IF(IFILE.LE.19) GO TO 912
C
      NCNST=N1
C
 1234 CONTINUE
      N1= NCNST
      N1=N1*4
      N=N1+NE1
      M=N1*6+NE1
      IXJAC=M
      LXJTJ=(N+1)*N/2
      LWORK=5*N+2*M+LXJTJ
C
C
C GET THE DATA POINTS
C READ DATA ELEMENTS WITHIN A DATA WINDOW
      K0=1+M3-NE1-1
      K1=M+M3-NE1-1
      DO 30 I=1,M
         K=I+M3-NE1-1
   30    Y(I)=X0(K)
C
C
      CALL FRTCMS('CLRSCRN ')
C
C     WRITE(6,163)
C 163 FORMAT(1X,'DO YOU WANT TO STORE THE DATA SET INTO THE FILE STORAGE
C    * - <Y/N>')
C     READ(5,51) IANS
C     IF(IANS.NE.IY) GO TO 3
C     WRITE(IFILE,164) K0,K1
C     WRITE(6,164) K0,K1
C 164 FORMAT(1X,//,15X,'TARGET AND FULL DATA POINTS OF A WINDOW',
C    */,             15X,'=======================================',
C    */,1X,'K=',I4,' TO K=',I4,/)
C     WRITE(IFILE,165) NT1,NT2,NW1,NW2,NC1,ND2,NF1,NF2
C     WRITE(6,165) NT1,NT2,NW1,NW2,NC1,NC2,NF1,NF2
C 165 FORMAT(1X,'TGT-TYPE : ',2A4,4X,'W/F-TYPE : ',2A4,4X,/,
C    */,     1X,'C/T-CATE : ',2A4,4X,'FILE-NUM : ',2A4,//,
C    *7X, 'X(K)',/)
C
C     DO 17 I=1,M
C        WRITE(6,166) Y(I)
```

```
C  17     WRITE(IFILE,166) Y(I)
C 166 FORMAT(1X,F14.8)
C
C GET THE INITIAL GUESS
    3 WRITE(6,63)
   63 FORMAT(1X,'DO YOU WANT TO INPUT THE INITIAL GUESSES USING A FILE?
     *- <Y/N>',/,1X, 'IF <N>, YOU ARE TO INPUT THEM INTERACTIVELY')
      READ(5,51) IANS
      IF(NE1.GT.0) GO TO 4
      DO 20  I=1,N
   20     X(I)=C.0
      IF(IANS.EQ.IY)  GO TO 4
      IF(IANS.EQ.IZ)  CALL INGU(X,N,N1,NE1)
      GO TO 5
    4 CALL DATA(X,N,N1,NE1)
    5 CONTINUE
C
C CALL  PROCESS ROUTINE
      CALL ZXSSQ(FUNC,M,N,NSIG,EPS,DELTA,MAXFN,ICPT,PARM,X,SSQ,F,
     *              XJAC,IXJAC,XJTJ,WORK,INFER,IER,T2,M3,N1,NE1)
C
C
C PUT THE RESULTS OUT
      CALL FRTCMS('CLRSCRN ')
      WRITE(IFILE,100) NE1
C
      WRITE(6,100) NE1
  100 FORMAT(1X,//,2X,'RESULTS OF OPTIMIZATION WITH ',I2,' EXTRA DATA PO
     *INTS',/,2X,'------- -- ------------ ---- -- ----- ---- ------',//)
C
      WRITE(IFILE,109) NT1,NT2,NW1,NW2,ND1,ND2,NF1,NF2
      WRITE(6,109) NT1,NT2,NW1,NW2,ND1,ND2,NF1,NF2
  109 FORMAT(2X,'TGT-TYPE : ',2A4,4X,'W/F-TYPE : ',2A4,4X,
     */,      2X,'C/T-DATE : ',2A4,4X,'FILE-NUM : ',2A4,//)
C
      WRITE(IFILE,101)
C
      WRITE(6,101)
  101 FORMAT(2X,'A. OPTIMIZED RESIDUES AND POLES',
     */,      2X,'-- --------- -------- --- -----',/,
     *2X,'PAIR #',6X,'RES.-REAL',6X,'RES.-IMAG',
     *              6X,'POLE-REAL',6X,'POLE-IMAG')
      IO=0
      DO 40  I=1,N1,4
          IO=IC+1
C
          WRITE(IFILE,102)IO,X(I),X(I+1),X(I+2),X(I+3)
C
```

```
      WRITE(6,102)IO,X(I),X(I+1),X(I+2),X(I+3)
102   FORMAT(2X,I4,3X,4(F14.8,1X))
 40 CONTINUE
    IF(NE1.EQ.0) GO TO 105
C
    WRITE(IFILE,103)
C
    WRITE(6,103)
103 FORMAT(1X,//,2X,'B. OPTIMIZED VALUE OF E(K)',
   */,      2X,'-- --------- ----- -- ----')
    IO=0
    DO 50 I=1,NE1
      IO=IC+1
      K=N1+I
C
      WRITE(IFILE,104) IO,X(K)
C
      WRITE(6,104) IO,X(K)
104   FORMAT(2X,'E(',I2,')=',F14.8)
 5C CONTINUE
    NE1=NE1+1
    DO 90 I=1,NE1
      X(N1+I)=0
 90 CONTINUE
    IF(NE1.GE.M3) GO TO 1001
    WRITE(6,1003)
1003 FORMAT(1X,'IF YOU READY TO GO, TYPE ANY CHATACTER')
    READ(5,51) IANS
    GO TO 1234
105 CONTINUE
C
C PUT TERMINATE
1001 WRITE(6,1000)
1000 FORMAT(1X,10X,'*** PROCESSING COMPLETED ***')
C
    STCP
    END
C
C
C
C   ================================================================
C   =  SUBROLTINE PARAS - DETERMINE THE NUMBER OF PARAMETERS       =
C   =                          A(N),B(N),SIGMA(N),FREQ(N)          =
C   =                          E(1)...E(NTO-1)                     =
C   =                   SPECIFY THE DATA PCINT AT WHICH E(K)       =
C   =                   ASSUMED TO BE ZERO(M3 PCINTS)              =
C   ================================================================
      SUBROUTINE PARAS(M3,N1,NE1)
```

```
C
      IMPLICIT REAL*8(A-H,O-Z)
      DATA IG/'G'/,IB/'B'/
C
    1 CALL FRTCMS('CLRSCRN ')
C
      WRITE(6,61)
      READ(5,*) N1
      WRITE(6,62)
      READ(5,*) NE1
      WRITE(6,63)
      READ(5,*) M3
   61 FORMAT(1X,'ENTER THE NUMBER OF POLES OF THE RESPONSE (# OF POLE)')
   62 FORMAT(1X,'HOW MANY EARLY TIME DATA POINTS DO YOU WANT TO HAVE',
     */,       1X,'IN YOUR CURRENT DATA WINDOW ?      : 0 UP TO 10(MAX)')
   63 FORMAT(1X,'SPECIFY K* POINT AT WHICH E(K) IS ASSUMED TO BE ZERO',
     */,       1X,'MATCH IT TO YOU K* OF YOUR DATA GENERATION ROUTINE  ')
C
C CHANGE IF NECESSARY
    2 WRITE(6,64)
   64 FORMAT(1X,'TYPE <B> TO GO BACK TO CHANGE THE ABOVE VARIABLES',
     */,       1X,'OTHERWISE TYPE <G> TO GO')
      READ(5,51) IANS
   51 FORMAT(A1)
      IF(IANS.EQ.IB) GO TO 1
      IF(IANS.NE.IG) GO TO 2
C
C
      RETURN
      END
C
C
C    ==================================================================
C    = SUBROUTINE INGU : INPUT THE INITIAL TRIALS FOR THE UNKNOWNS     =
C    ==================================================================
C
      SUBROUTINE INGU(X,N,N1,NE1)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DATA IY/'Y'/,IZ/'N'/
      DIMENSION X(N)
C
    1 CALL FRTCMS('CLRSCRN ')
C
      WRITE(6,61)
   61 FORMAT(1X,'ENTER THE INITIAL VALUE OF THE X(K) PARAMETERS')
      IO=0
      DO 10 I=1,N1,4
```

```
      IO=IC+1
      WRITE(6,611) IO,IO,IO,IO
611   FORMAT(1X,'R -RE(',I2,')=',2X,'R-IM(',I2,')=',
     *       2X,'SIGMA(',I2,')=',2X,'FREQ(',I2,')=')
 10   READ(5,*) X(I),X(I+1),X(I+2),X(I+3)
C
C
      IF(NE1.EQ.0) GO TO 2
C
C INITIAL GUESS FOR THE ERROR PARAMETER
      DO 20 J=1,NE1
      WRITE(6,612) J
612   FORMAT(1X,'E(',I2,')=')
      K=N1+J
 20   READ(5,*) X(K)
C
    2 WRITE(6,62)
   62 FORMAT(1X,'DO YOU NEED TO CHANGE THE ABOVE VALUES?- <Y/N>')
      READ(5,51) IANS
   51 FORMAT(A1)
      IF(IANS.EQ.IY) GOTO 1
      IF(IANS.EQ.IZ) GOTO 3
      GOTO 2
    3 CONTINUE
      RETURN
      END
C
C
C     ==================================================================
C     = SUBROUTINE FUNC :  SUPPLY THE EVALUATED OBJECT FUNCTION VALUE   =
C     =                    EXTERNALLY.                                  =
C     ==================================================================
C
      SUBROUTINE FUNC(X,M,N,F,T2,M3,N1,NE1)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(N),F(M),Y(110),V(110)
      COMMON /ZSQ/Y,V
C
C USER DEFINC FUCTION AREA
C ========================
      PI=3.141592654
C
      XT=0.0
      TO=T2/511.
      LOCP=M
      DO 10 I=1,LOOP
         ICOUNT=I+M3-NE1-2
```

```
      V(I)=T0*ICOUNT
      IF(X(1).LE.10D-8) X(1)=10D-8
      IF(X(5).LE.10D-8) X(5)=10D-8
      IF(X(9).LE.10D-8) X(9)=10D-8
      IF(X(13).LE.10D-8) X(13)=10D-8
      IF(X(17).LE.10D-8) X(17)=10D-8
      IF(X(21).LE.10D-8) X(21)=10D-8
      FF1=X(2)/X(1)
      FF2=X(6)/X(5)
      FF3=X(10)/X(9)
      FF4=X(14)/X(13)
      FF5=X(18)/X(17)
      FF6=X(22)/X(21)
C
      F(I)=Y(I)- DSQRT(X( 1)**2 + X( 2)**2) * DEXP(X( 3)*V(I))
     *          * DCOS(2*PI*X( 4)*V(I) + DATAN(FF1))
     *          - DSQRT(X( 5)**2 + X( 6)**2) * DEXP(X( 7)*V(I))
     *          * DCOS(2*PI*X( 8)*V(I) + DATAN(FF2))
     *          - DSQRT(X( 9)**2 + X(10)**2) * DEXP(X(11)*V(I))
     *          * DCOS(2*PI*X(12)*V(I) + DATAN(FF3))
     *          - DSQRT(X(13)**2 + X(14)**2) * DEXP(X(15)*V(I))
     *          * DCOS(2*PI*X(16)*V(I) + DATAN(FF4))
     *          - DSQRT(X(17)**2 + X(18)**2) * DEXP(X(19)*V(I))
     *          * DCOS(2*PI*X(20)*V(I) + DATAN(FF5))
     *          - DSQRT(X(21)**2 + X(22)**2) * DEXP(X(23)*V(I))
     *          * DCOS(2*PI*X(24)*V(I) + DATAN(FF6))
      IF(NE1.LT.0) GO TO 10
      IF(I.GT.NE1) GO TO 10
C USER DEFINED NON-LINEAR FUNCTION AREA
C ARRAY
         K=N-I+1
         F(I)=F(I)- X(K)
   10 CONTINUE
C END OF USER DEFINED FUNCTION AREA
C
C
      RETURN
      END
C
C
C   =================================================================
C   PURPOSE                - MINIMUM OF THE SUM OF SQUARES OF M FUNCTIONS
C                            IN N  VARIABLES  USING  A  FINITE DIFFERENCE
C                            LEVENBERG-MARQUARDT ALGORITHM
C   =================================================================
      SUBROUTINE ZXSSQ  (FUNC,M,N,NSIG,EPS,DELTA,MAXFN,IOPT,PARM,
     *                   X,SSQ,F,XJAC,IXJAC,XJTJ,WORK,INFER,IER,I2,M3,
     *                   N1,NE1)
```

```
C
      INTEGER            M,N,NSIG,MAXFN,IOPT,IXJAC,INFER,IER,M3
      DOUBLE PRECISION   EPS,DELTA,PARM(1),X(N),SSQ,F(M),XJAC(1),
     *                   XJTJ(1),WORK(1),T2
C                                  XJAC USED INTERNALLY IN PACKED FORM
C                                  SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER            IMJC,IGRAD1,IGRADL,IGRADU,IDELX1,IDELXL,
     *                   IDELXU,ISCAL1,ISCALL,ISCALU,IXNEW1,IXNEWL,
     *                   IXBAD1,IFPL1,IFPL,IFPU,IFML1,IFML,IEVAL,
     *                   IBAD,ISW,ITER,J,IJAC,I,K,L,IS,JS,LI,LJ,ICOUNT,
     *                   IZERO,LEVEL,LEVOLD
      DOUBLE PRECISION   AL,CONS2,DNORM,DSQ,
     *                   ERL2,ERL2X,F0,F0SQ,F0SQS4,G,HALF,
     *                   HH,ONE,ONEP10,ONEP5,ONESF0,AX,
     *                   PREC,REL,RHH,SIG,SQDIF,SSQOLD,SUM,TEN,
     *                   TENTH,XDIF,XHOLD,UP,ZERO,
     *                   XDABS,RELCON,P01,TWO,HUNTW,DELTA2
      DATA               SIG/16.0D0/
      DATA               AX/0.1D0/
      DATA               P01,TENTH,HALF,ZERO,ONE,ONEP5,TWO,
     *                   TEN,HUNTW,ONEP10/0.01D0,0.1D0,0.5D0,0.0D0,
     *                   1.D0,1.5D0,2.D0,10.0D0,1.2D2,1.D10/
C                                  ERROR CHECKS
C                                  FIRST EXECUTABLE STATEMENT
C
C
      IER = 0
      LEVEL = 0
      CALL UERSET (LEVEL,LEVOLD)
      IF (M.LE.0.OR.M.GT.IXJAC.OR.N.LE.0.OR.IOPT.LT.0.OR.IOPT.GT.2)
     *   GO TO 305
      IMJC = IXJAC-M
      IF (IOPT.NE.2) GO TO 5
      IF (PARM(2).LE.ONE.OR.PARM(1).LE.ZERO) GO TO 305
C                                  MACHINE DEPENDENT CONSTANTS
    5 PREC = TEN**(-SIG-ONE)
      REL = TEN**(-SIG*HALF)
      RELCON = TEN**(-NSIG)
C                                  WORK VECTOR IS CONCATENATION OF
C                                  SCALED HESSIAN,GRADIENT,DELX,SCALE,
C                                  XNEW,XBAD,F(X+DEL),F(X-DEL)
      IGRAD1 = ((N+1)*N)/2
      IGRADL = IGRAD1+1
      IGRADU = IGRAD1+N
      IDELX1 = IGRADU
      IDELXL = IDELX1+1
      IDELXU = IDELX1+N
      ISCAL1 = IDELXU
```

```
      ISCALL = ISCAL1+1
      ISCALU = ISCAL1+N
      IXNEW1 = ISCALU
      IXNEWL = IXNEW1+1
      IXBAD1 = IXNEW1+N
      IFPL1 = IXBAD1+N
      IFPL = IFPL1+1
      IFPU = IFPL1+M
      IFML1 = IFPU
      IFML = IFML1+1
      IMJC = IXJAC - M
C                                  INITIALIZE VARIABLES
      AL = ONE
      CONS2 = TENTH
      IF (IOPT.EQ.0) GO TO 20
      IF (IOPT.EQ.1) GO TO 10
      AL = PARM(1)
      FO = PARM(2)
      UP = PARM(3)
      CONS2 = PARM(4)*HALF
      GO TO 15
   10 AL = P01
      FO = TWO
      UP = HUNTH
   15 ONESFO = ONE/FO
      FOSQ = FO*FO
      FOSQS4 = FOSQ**4
   20 IEVAL = 0
      DELTA2 = DELTA*HALF
      ERL2 = ONEP10
      IBAD = -99
      ISW = 1
      ITER = -1
      INFER = 0
      IER = 0
      DO 25 J=IDELXL,IDELXU
         WORK(J) = ZERO
   25 CONTINUE
      GO TO 165
C                                  MAIN LOOP
   30 SSQOLD = SSQ
C                                  CALCULATE JACOBIAN
      IF (INFER.GT.0.OR.IJAC.GE.N.OR.IOPT.EQ.0.OR.ICOUNT.GT.0) GO TO 55
C                                  RANK ONE UPDATE TO JACOBIAN
      IJAC = IJAC+1
      DSQ = ZERO
      DO 35 J=IDELXL,IDELXU
         DSQ = DSQ+WORK(J)*WORK(J)
```

```
   35 CONTINUE                                                          PRO04810
      IF (DSC.LE.ZERO) GO TO 55                                         PRO04820
      DO 50 I=1,M                                                       PRO04830
         G = F(I)-WORK(IFML1+I)                                         PRO04840
         K = I                                                          PRO04850
         DO 40 J=IDELXL,IDELXU                                          PRO04860
            G = G+XJAC(K)*WORK(J)                                       PRO04870
            K = K+IXJAC                                                 PRO04880
   40    CONTINUE                                                       PRO04890
         G = G/DSC                                                      PRO04900
         K = I                                                          PRO04910
         DO 45 J=IDELXL,IDELXU                                          PRO04920
            XJAC(K) = XJAC(K)-G*WORK(J)                                 PRO04930
            K = K+IXJAC                                                 PRO04940
   45    CONTINUE                                                       PRO04950
   50 CONTINUE                                                          PRO04960
      GO TO 80                                                          PRO04970
C                                 JACOBIAN BY INCREMENTING X            PRO04980
   55 IJAC = 0                                                          PRO04990
      K = -IMJC                                                         PRO05000
      DO 75 J=1,N                                                       PRO05010
         K = K+IMJC                                                     PRO05020
         XDABS = DABS(X(J))                                             PRO05030
         HH = REL*(DMAX1(XDABS,AX))                                     PRO05040
         XHOLD = X(J)                                                   PRO05050
         X(J) = X(J)+HH                                                 PRO05060
C                                                                       PRO05070
C CALL FUNC                                                             PRO05080
         CALL FUNC (X,M,N,WORK(IFPL),T2,M3,N1,NE1)                      PRO05090
         IEVAL = IEVAL+1                                                PRO05100
         X(J) = XHOLD                                                   PRO05110
         IF (ISW.EQ.1) GO TO 65                                         PRO05120
C                                 CENTRAL DIFFERENCES                   PRO05130
         X(J) = XHOLD-HH                                                PRO05140
C                                                                       PRO05150
C CALL FUNC                                                             PRO05160
         CALL FUNC (X,M,N,WORK(IFML),T2,M3,N1,NE1)                      PRO05170
         IEVAL = IEVAL+1                                                PRO05180
         X(J) = XHOLD                                                   PRO05190
         RHH = HALF/HH                                                  PRO05200
         DO 60 I=IFPL,IFPU                                              PRO05210
            K = K+1                                                     PRO05220
            XJAC(K) = (WORK(I)-WORK(I+M))*RHH                           PRO05230
   60    CONTINUE                                                       PRO05240
         GO TO 75                                                       PRO05250
C                                 FORWARD DIFFERENCES                   PRO05260
   65    RHH = ONE/HH                                                   PRO05270
         DO 70 I=1,M                                                    PRO05280
```

```
            K = K+1
            XJAC(K) = (WORK(IFPL1+I)-F(I))*RHH
   70    CONTINUE
   75 CONTINUE
C                                        CALCULATE GRADIENT
   80 ERL2X = ERL2
      ERL2 = ZERO
      K = -IMJC
      DO 90 J=IGRADL,IGRADU
         K = K+IMJC
         SUM = ZERO
         DO 85 I=1,M
            K = K+1
            SUM = SUM+XJAC(K)*F(I)
   85    CONTINUE
         WORK(J) = SUM
         ERL2 = ERL2+SUM*SUM
   90 CONTINUE
      ERL2 = DSQRT(ERL2)
C                                        CONVERGENCE TEST FOR NORM OF GRADIENT
      IF (IJAC.GT.0) GO TO 95
      IF (ERL2.LE.DELTA2) INFER = INFER+4
      IF (ERL2.LE.CONS2) ISW = 2
C                                        CALCULATE THE LOWER SUPER TRIANGE OF
C                                        JACOBIAN (TRANSPOSED) * JACOBIAN
   95 L = 0
      IS = -IXJAC
      DO 110 I=1,N
         IS = IS+IXJAC
         JS = -IXJAC
         DO 105 J=1,I
            JS = JS+IXJAC
            L = L+1
            SUM = ZERO
            DO 100 K=1,M
               LI = IS+K
               LJ = JS+K
               SUM = SUM+XJAC(LI)*XJAC(LJ)
  100       CONTINUE
            XJTJ(L) = SUM
  105    CONTINUE
  110 CONTINUE
C                                        CONVERGENCE CHECKS
      IF (INFER.GT.0) GO TO 315
      IF (IEVAL.GE.MAXFN) GO TO 290
C                                        COMPUTE SCALING VECTOR
      IF (IGFT.EQ.0) GO TO 120
      K = 0
```

```
      DO 115 J=1,N
         K = K+J
         WORK(ISCAL1+J) = XJTJ(K)
  115 CONTINUE
      GO TO 135
C                                      COMPUTE SCALING VECTOR AND NORM
  120 DNORM = ZERO
      K = 0
      DO 125 J=1,N
         K = K+J
         WORK(ISCAL1+J) = DSQRT(XJTJ(K))
         DNORM = DNORM+XJTJ(K)*XJTJ(K)
  125 CONTINUE
      DNORM = ONE/DSQRT(DNORM)
C                                      NORMALIZE SCALING VECTOR
      DO 130 J=ISCALL,ISCALU
         WORK(J) = WORK(J)*DNORM*ERL2
  130 CONTINUE
C                                      ADD L-M FACTOR TO DIAGONAL
  135 ICOUNT = 0
  140 K = 0
      DO 150 I=1,N
         DO 145 J=1,I
            K = K+1
            WORK(K) = XJTJ(K)
  145    CONTINUE
         WORK(K) = WORK(K)+WORK(ISCAL1+I)*AL
         WORK(IDELX1+I) = WORK(IGRAD1+I)
  150 CONTINUE
C                                      CHOLESKY DECOMPOSITION
  155 CALL LEQT1P (WORK,1,N,WORK(IDELXL),N,0,G,XHOLD,IER)
      IF (IER.EQ.0) GO TO 160
      IER = 0
      IF (IJAC.GT.0) GO TO 55
      IF (IBAD.LE.0) GO TO 240
      IF (IBAD.GE.2) GO TO 310
      GO TO 190
  160 IF (IBAD.NE.-99) IBAD = 0
C                                      CALCULATE SUM OF SQUARES
  165 DO 170 J=1,N
         WORK(IXNEW1+J) = X(J)-WORK(IDELX1+J)
  170 CONTINUE
C
C CALL FUNC
      CALL FUNC (WORK(IXNEWL),M,N,WORK(IFPL),T2,M3,N1,NE1)
      IEVAL = IEVAL+1
      SSQ = ZERO
      DO 175 I=IFPL,IFPU
```

```
      SSQ = SSC+WORK(I)*WORK(I)
  175 CONTINUE
      IF (ITER.GE.0) GO TO 185
C                                      SSQ FOR INITIAL ESTIMATES OF X
      ITER = C
      SSCOLD = SSQ
      DO 180 I=1,M
         F(I) = WCRK(IFPL1+I)
  18C CONTINUE
      GO TO 55
  185 IF (IOPT.EQ.0) GO TO 215
C                                      CHECK DESCENT PROPERTY
      IF (SSQ.LE.SSQOLD) GO TO 205
C                                      INCREASE PARAMETER AND TRY AGAIN
  19C ICOUNT = ICCUNT+1
      AL = AL*FOSC
      IF (IJAC.EQ.0) GC TO 195
      IF (ICCUNT.GE.4.OR.AL.GT.UP) GC TO 200
  195 IF (AL.LE.UP) GC TO 140
      IF (IBAC.EQ.1) GO TC 310
      IER = 39
      GO TO 315
  200 AL = AL/FOSCS4
      GO TO 55
C                                      ADJUST MARQUARDT PARAMETER
  205 IF (ICCUNT.EQ.0) AL = AL/FO
      IF (ERL2X.LE.ZERO) GO TO 210
      G = ERL2/ERL2X
      IF (ERL2.LT.ERL2X) AL = AL*DMAX1(ONESFO,G)
      IF (ERL2.GT.ERL2X) AL = AL*DMIN1(FO,G)
  210 AL = DMAX1(AL,PREC)
C                                      ONE ITERATION CYCLE COMPLETED
  215 ITER = ITER+1
      DO 220 J=1,N
         X(J) = WORK(IXNEW1+J)
  22C CONTINUE
      DO 225 I=1,M
         WORK(IFML1+I) = F(I)
         F(I) = WCRK(IFPL1+I)
  225 CONTINUE
C                                      RELATIVE CONVERGENCE TEST FOR X
      IF (AL.GT.5.0D0) GO TO 30
      DO 230 J=1,N
         XDIF = DABS(WORK(IDELX1+J))/DMAX1(DABS(X(J)),AX)
         IF (XCIF.GT.RELCON) GO TO 235
  23C CONTINUE
      INFER = 1
C                                      RELATIVE CONVERGENCE TEST FOR SSQ
```

```
  235 SQDIF = DABS(SSQ-SSQOLD)/DMAX1(SSQOLD,AX)
      IF (SQDIF.LE.EPS) INFER = INFER+2
      GO TO 30
C                                     SINGULAR DECOMPOSITION
  240 IF (IBAD) 255,245,265
C                                     CHECK TO SEE IF CURRENT
C                                     ITERATE HAS CYCLED BACK TO
C                                     THE LAST SINGULAR POINT
  245 DO 250 J=1,N
         XHOLD = WORK(IXBAD1+J)
         IF (DABS(X(J)-XHOLD).GT.RELCON*DMAX1(AX,DABS(XHOLD))) GO TO 255
  250 CONTINUE
      GO TO 295
C                                     UPDATE THE BAD X VALUES
  255 DO 260 J=1,N
         WORK(IXBAD1+J) = X(J)
  260 CONTINUE
      IBAD = 1
C                                     INCREASE DIAGONAL OF HESSIAN
  265 IF (IOPT.NE.0) GO TO 280
      K = 0
      DO 275 I=1,N
         DO 270 J=1,I
            K = K+1
            WORK(K) = XJTJ(K)
  270    CONTINUE
         WORK(K) = ONEP5*(XJTJ(K)+AL*ERL2*WORK(ISCAL1+I))+REL
  275 CONTINUE
      IBAD = 2
      GO TO 155
C                                     REPLACE ZEROES ON HESSIAN DIAGONAL
  280 IZERO = 0
      DO 285 J=ISCALL,ISCALU
         IF (WORK(J).GT.ZERO) GO TO 285
         IZERO = IZERO+1
         WORK(J) = ONE
  285 CONTINUE
      IF (IZERO.LT.N) GO TO 140
      IER = 38
      GO TO 315
C                                     TERMINAL ERROR
  290 IER = IER+1
  295 IER = IER+1
      IER = IER+1
  305 IER = IER+1
  310 IER = IER+129
      IF (IER.EQ.130) GO TO 335
C                                     OUTPUT ERL2,IEVAL,NSIG,AL, AND ITER
```

```
  315 G = SIG
      DO 320 J=1,N
         XHOLD = DABS(WORK(IDELX1+J))
         IF (XHOLD.LE.ZERO) GO TO 320
         G = DMIN1(G,-DLOG10(XHOLD)+DLOG10(DMAX1(AX,DABS(X(J)))))
  320 CONTINUE
      IF(N.GT.2) GO TO 330
      DO 325 J = 1,N
  325 WORK(J+5) = WORK(J+IGRAD1)
  330 WORK(1) = ERL2+ERL2
      WORK(2) = IEVAL
      SSQ = SSQOLD
      WORK(3) = G
      WORK(4) = AL
      WORK(5) = ITER
  335 CALL UERSET(LEVOLD,LEVOLD)
      IF (IER.EQ.0) GO TO 9005
 9000 CONTINUE
      CALL UERTST (IER,6HZXSSQ )
 9005 RETURN
      END
C
C    ==========================================================================
C    SUBROUTINE LEQT1P
C    PURPOSE             - LINEAR EQUATION SOLUTION - POSITIVE DEFINITE
C                            MATRIX - SYMMETRIC STORAGE MODE - SPACE
C                            ECONOMIZER SOLUTION
C    ==========================================================================
C
      SUBROUTINE LEQT1P (A,M,N,B,IB,IDGT,D1,D2,IER)
C
      DIMENSION          A(1),B(IB,1)
      DOUBLE PRECISION   A,B,D1,D2
C                                  FIRST EXECUTABLE STATEMENT
C                                  INITIALIZE IER
      IER = 0
C                                  DECOMPOSE A
      CALL LUDECP (A,A,N,D1,D2,IER)
      IF (IER.NE.0) GO TO 9000
C                                  PERFORM ELIMINATION
      DO 5 I = 1,M
         CALL LUELMP (A,B(1,I),N,B(1,I))
    5 CONTINUE
      GO TO 9005
 9000 CONTINUE
      CALL UERTST(IER,6HLEQT1P)
 9005 RETURN
      END
```

```
  315 G = SIG
      DO 320 J=1,N
         XHOLD = DABS(WORK(IDELX1+J))
         IF (XHOLD.LE.ZERO) GO TO 320
         G = DMIN1(G,-DLOG10(XHOLD)+DLOG10(DMAX1(AX,DABS(X(J)))))
  320 CONTINUE
      IF(N.GT.2) GO TO 330
      DO 325 J = 1,N
  325 WORK(J+5) = WORK(J+IGRAD1)
  330 WORK(1) = ERL2+ERL2
      WORK(2) = IEVAL
      SSQ = SSQOLD
      WORK(3) = G
      WORK(4) = AL
      WORK(5) = ITER
  335 CALL UERSET(LEVOLD,LEVOLD)
      IF (IER.EQ.0) GO TO 9005
 9000 CONTINUE
      CALL UERTST (IER,6HZXSSQ )
 9005 RETURN
      END
C
C   ===================================================================
C   SUBROUTINE LEQT1P
C   PURPOSE             - LINEAR EQUATION SOLUTION - POSITIVE DEFINITE
C                           MATRIX - SYMMETRIC STORAGE MODE - SPACE
C                           ECONOMIZER SOLUTION
C   ===================================================================
C
      SUBROUTINE LEQT1P (A,M,N,B,IB,IDGT,D1,D2,IER)
C
      DIMENSION          A(1),B(IB,1)
      DOUBLE PRECISION   A,B,D1,D2
C                                  FIRST EXECUTABLE STATEMENT
C                                  INITIALIZE IER
      IER = 0
C                                  DECOMPOSE A
      CALL LUDECP (A,A,N,D1,D2,IER)
      IF (IER.NE.0) GO TO 9000
C                                  PERFORM ELIMINATION
      DO 5 I = 1,M
         CALL LUELMP (A,B(1,I),N,B(1,I))
    5 CONTINUE
      GO TO 9005
 9000 CONTINUE
      CALL UERTST(IER,6HLEQT1P)
 9005 RETURN
      END
```

```
C                                                                        PR007690
C     ===================================================================PR007700
C     SUBROUTINE LUDECP                                                  PR007710
C     PURPOSE              - DECOMPOSITION OF A POSITIVE DEFINITE MATRIX - PR007720
C                            SYMMETRIC STORAGE MODE                      PR007730
C     ===================================================================PR007740
C                                                                        PR007750
      SUBROUTINE LUDECP (A,UL,N,D1,D2,IER)                               PR007760
C                                                                        PR007770
      DIMENSION          A(1),UL(1)                                      PR007780
      DOUBLE PRECISION   A,UL,D1,D2,ZERO,ONE,FOUR,SIXTN,SIXTH,X,RN       PR007790
      DATA               ZERO,ONE,FOUR,SIXTN,SIXTH/                      PR007800
     *                   0.0D0,1.D0,4.D0,16.D0,.0625D0/                  PR007810
C                                   FIRST EXECUTABLE STATEMENT           PR007820
      D1=ONE                                                             PR007830
      D2=ZERO                                                            PR007840
      RN = ONE/(N*SIXTN)                                                 PR007850
      IP = 1                                                             PR007860
      IER=0                                                              PR007870
      DO 45  I = 1,N                                                     PR007880
         IQ = IP                                                         PR007890
         IR = 1                                                          PR007900
         DO 40  J = 1,I                                                  PR007910
            X = A(IP)                                                    PR007920
            IF (J .EQ. 1) GO TO 10                                       PR007930
            DO 5  K=IQ,IP1                                               PR007940
               X = X - UL(K) * UL(IR)                                    PR007950
               IR = IR+1                                                 PR007960
    5       CONTINUE                                                     PR007970
   10       IF (I.NE.J) GO TO 30                                         PR007980
            D1 = D1*X                                                    PR007990
            IF (A(IP) + X*RN .LE. A(IP)) GO TO 50                        PR008000
   15       IF (DABS(D1).LE.ONE) GO TO 20                                PR008010
            D1 = D1 * SIXTH                                              PR008020
            D2 = D2 + FOUR                                               PR008030
            GO TO 15                                                     PR008040
   20       IF (DABS(D1) .GE. SIXTH) GO TO 25                            PR008050
            D1 = D1 * SIXTN                                              PR008060
            D2 = D2 - FOUR                                               PR008070
            GO TO 20                                                     PR008080
   25       UL(IP) = ONE/DSQRT(X)                                        PR008090
            GO TO 35                                                     PR008100
   30       UL(IP) = X * UL(IR)                                          PR008110
   35       IP1 = IP                                                     PR008120
            IP = IP+1                                                    PR008130
            IR = IR+1                                                    PR008140
   40    CONTINUE                                                        PR008150
   45 CONTINUE                                                           PR008160
```

```
      GO TO 9005
   50 IER = 129
 9000 CONTINUE
      CALL UERTST(IER,6HLUDECP)
 9005 RETURN
      END
C
C   ==================================================================
C   PURPOSE              - ELIMINATION PART OF THE SOLUTION OF AX=B -
C                          POSITIVE DEFINITE MATRIX - SYMMETRIC
C                          STORAGE MODE
C   ==================================================================
C
      SUBROUTINE LUELMP (A,B,N,X)
C
      DIMENSION          A(1),B(1),X(1)
      DOUBLE PRECISION   A,B,X,T,ZERO
      DATA               ZERO/0.0D0/
C                                  FIRST EXECUTABLE STATEMENT
C                                  SOLUTION OF LY = B
      IP=1
      IW = 0
      DO 15 I=1,N
         T=B(I)
         IM1 = I-1
         IF (IW .EQ. 0) GO TO 9
         IP=IP+IW-1
         DO 5 K=IW,IM1
            T = T-A(IP)*X(K)
            IP=IP+1
    5    CONTINUE
         GO TO 10
    9    IF (T .NE. ZERO) IW = I
         IP = IP+IM1
   10    X(I)=T*A(IP)
         IP=IP+1
   15 CONTINUE
C                                  SOLUTION OF UX = Y
      N1 = N+1
      DO 30 I = 1,N
         II = N1-I
         IP=IP-1
         IS=IP
         IQ=II+1
         T=X(II)
         IF (N.LT.IQ) GO TO 25
         KK = N
         DO 20 K=IQ,N
```

```
          T = T - A(IS) * X(KK)
          KK = KK-1
          IS = IS-KK
   20     CONTINUE
   25     X(II)=T*A(IS)
   30  CONTINUE
       RETURN
       END
C
C   ===========================================================================
C   SUBROUTINE UERSET
C   PURPOSE               - SET MESSAGE LEVEL FOR IMSL ROUTINE UERTST
C   ===========================================================================
C   MSG :                       LEVEL = 4 CAUSES ALL MESSAGES TO BE
C                                         PRINTED,
C                               LEVEL = 3 MESSAGES ARE PRINTED IF IER IS
C                                         GREATER THAN 32,
C                               LEVEL = 2 MESSAGES ARE PRINTED IF IER IS
C                                         GREATER THAN 64,
C                               LEVEL = 1 MESSAGES ARE PRINTED IF IER IS
C                                         GREATER THAN 128,
C                               LEVEL = 0 ALL MESSAGE PRINTING IS
C                                         SUPPRESSED.
C
      SUBROUTINE UERSET (LEVEL,LEVOLD)
C                                  SPECIFICATIONS FOR ARGUMENTS
      INTEGER            LEVEL,LEVOLD
C                                  FIRST EXECUTABLE STATEMENT
      LEVOLD = LEVEL
      CALL UERTST (LEVOLD,6HUERSET)
      RETURN
      END
C
C   ===========================================================================
C   SUBROUTINE UERTST
C   PURPOSE               - PRINT A MESSAGE REFLECTING AN ERROR CONDITION
C   ===========================================================================
C   ARGUMENTS :   IER     - ERROR PARAMETER. (INPUT)
C                           IER = I+J WHERE
C                           I = 128 IMPLIES TERMINAL ERROR MESSAGE,
C                           I =  64 IMPLIES WARNING WITH FIX MESSAGE,
C                           I =  32 IMPLIES WARNING MESSAGE.
C                           J = ERROR CODE RELEVANT TO CALLING
C                                 ROUTINE.
C                 NAME    - A CHARACTER STRING OF LENGTH SIX PROVIDING
C                             THE NAME OF THE CALLING ROUTINE. (INPUT)
C
      SUBROUTINE UERTST (IER,NAME)
```

```
C                                     SPECIFICATIONS FOR ARGUMENTS
      INTEGER            IER
      INTEGER            NAME(1)
C                                     SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER            I,IEQ,IEQDF,IOUNIT,LEVEL,LEVOLD,NAMEQ(6),
     *                   NAMSET(6),NAMUPK(6),NIN,NMTB
      DATA               NAMSET/1HU,1HE,1HR,1HS,1HE,1HT/
      DATA               NAMEQ/6*1H /
      DATA               LEVEL/4/,IEQDF/0/,IEQ/1H=/
C                                     UNPACK NAME INTO NAMUPK
C                                     FIRST EXECUTABLE STATEMENT
      CALL USPKD (NAME,6,NAMUPK,NMTB)
C                                     GET OUTPUT UNIT NUMBER
      CALL UGETIO(1,NIN,IOUNIT)
C                                     CHECK IER
      IF (IER.GT.999) GO TO 25
      IF (IER.LT.-32) GO TO 55
      IF (IER.LE.128) GO TO 5
      IF (LEVEL.LT.1) GO TO 30
C                                     PRINT TERMINAL MESSAGE
      IF (IEQDF.EQ.1) WRITE(IOUNIT,35) IER,NAMEQ,IEQ,NAMUPK
      IF (IEQDF.EQ.0) WRITE(IOUNIT,35) IER,NAMUPK
      GO TO 30
    5 IF (IER.LE.64) GO TO 10
      IF (LEVEL.LT.2) GO TO 30
C                                     PRINT WARNING WITH FIX MESSAGE
      IF (IEQDF.EQ.1) WRITE(IOUNIT,40) IER,NAMEQ,IEQ,NAMUPK
      IF (IEQDF.EQ.0) WRITE(IOUNIT,40) IER,NAMUPK
      GO TO 30
   10 IF (IER.LE.32) GO TO 15
C                                     PRINT WARNING MESSAGE
      IF (LEVEL.LT.3) GO TO 30
      IF (IEQDF.EQ.1) WRITE(IOUNIT,45) IER,NAMEQ,IEQ,NAMUPK
      IF (IEQDF.EQ.0) WRITE(IOUNIT,45) IER,NAMUPK
      GO TO 30
   15 CONTINUE
C                                     CHECK FOR UERSET CALL
      DO 20 I=1,6
         IF (NAMUPK(I).NE.NAMSET(I)) GO TO 25
   20 CONTINUE
      LEVOLD = LEVEL
      LEVEL = IER
      IER = LEVOLD
      IF (LEVEL.LT.0) LEVEL = 4
      IF (LEVEL.GT.4) LEVEL = 4
      GO TO 30
   25 CONTINUE
      IF (LEVEL.LT.4) GO TO 30
```

```
C                                  PRINT NON-DEFINED MESSAGE            PRO09610
      IF (IEQDF.EQ.1) WRITE(IOUNIT,50) IER,NAMEQ,IEQ,NAMUPK             PRO09620
      IF (IEQDF.EQ.0) WRITE(IOUNIT,50) IER,NAMUPK                       PRO09630
   30 IEQDF = 0                                                         PRO09640
      RETURN                                                            PRO09650
   35 FORMAT(19H *** TERMINAL ERROR,10X,7H(IER = ,I3,                   PRO09660
     1       20H) FROM IMSL ROUTINE ,6A1,A1,6A1)                        PRO09670
   40 FORMAT(27H *** WARNING WITH FIX ERROR,2X,7H(IER = ,I3,            PRO09680
     1       20H) FROM IMSL ROUTINE ,6A1,A1,6A1)                        PRO09690
   45 FORMAT(18H *** WARNING ERROR,11X,7H(IER = ,I3,                    PRO09700
     1       20H) FROM IMSL ROUTINE ,6A1,A1,6A1)                        PRO09710
   50 FORMAT(20H *** UNDEFINED ERROR,9X,7H(IER = ,I5,                   PRO09720
     1       20H) FROM IMSL ROUTINE ,6A1,A1,6A1)                        PRO09730
C                                                                       PRO09740
C                                  SAVE P FOR P = R CASE                PRO09750
C                                    P IS THE PAGE NAMUPK               PRO09760
C                                    R IS THE ROUTINE NAMUPK            PRO09770
   55 IEQDF = 1                                                         PRO09780
      DO 60 I=1,6                                                       PRO09790
   60 NAMEQ(I) = NAMUPK(I)                                              PRO09800
   65 RETURN                                                            PRO09810
      END                                                               PRO09820
C                                                                       PRO09830
C   ====================================================================PRO09840
C   SUBROUTINE UGETIO                                                   PRO09850
C   PURPOSE             - TO RETRIEVE CURRENT VALUES AND TO SET NEW     PRO09860
C                         VALUES FOR INPUT AND OUTPUT UNIT IDENTIFIERS. PRO09870
C   ====================================================================PRO09880
C   ARGUMENTS    IOPT   - OPTION PARAMETER. (INPUT)                     PRO09890
C                       1.IF IOPT=1, THE CURRENT INPUT AND OUTPUT       PRO09900
C                           UNIT IDENTIFIER VALUES ARE RETURNED IN NIN  PRO09910
C                           AND NOUT, RESPECTIVELY.                     PRO09920
C                       2.IF IOPT=2, THE INTERNAL VALUE OF NIN IS       PRO09930
C                           RESET FOR SUBSEQUENT USE.                   PRO09940
C                       3.IF IOPT=3, THE INTERNAL VALUE OF NOUT IS      PRO09950
C                           RESET FOR SUBSEQUENT USE.                   PRO09960
C                NIN    - INPUT UNIT IDENTIFIER.                        PRO09970
C                           OUTPUT IF IOPT=1, INPUT IF IOPT=2.          PRO09980
C                NOUT   - OUTPUT UNIT IDENTIFIER.                       PRO09990
C                           OUTPUT IF IOPT=1, INPLT IF IOPT=3.          PRO10000
C                                                                       PRO10010
      SUBROUTINE UGETIO(ICPT,NIN,NOUT)                                  PRO10020
C                                  SPECIFICATIONS FOR ARGUMENTS         PRO10030
      INTEGER            IOPT,NIN,NOUT                                  PRO10040
C                                  SPECIFICATIONS FOR LOCAL VARIABLES   PRO10050
      INTEGER            NIND,NOUTD                                     PRO10060
      DATA               NIND/5/,NOUTD/6/                               PRO10070
C                                  FIRST EXECUTABLE STATEMENT           PRO10080
```

```
      IF (IOPT.EQ.3) GO TO 10
      IF (IOPT.EQ.2) GO TO 5
      IF (IOPT.NE.1) GO TO 9005
      NIN = NIND
      NOUT = NOUTD
      GO TO 9005
    5 NIND = NIN
      GO TO 9005
   10 NOUTD = NOUT
 9005 RETURN
      END
C
C   ==================================================================
C   SUBROUTINE USCKED
C   PURPOSE            - NUCLEUS CALLED BY IMSL ROUTINES THAT HAVE
C                        CHARACTER STRING ARGUMENTS
C   ==================================================================
C   ARGUMENTS   PACKED - CHARACTER STRING TO BE UNPACKED.(INPUT)
C               NCHARS - LENGTH OF PACKED. (INPUT)  SEE REMARKS.
C               UNPAKD - INTEGER ARRAY TO RECEIVE THE UNPACKED
C                        REPRESENTATION OF THE STRING. (OUTPUT)
C               NCHMTB - NCHARS MINUS TRAILING BLANKS. (OUTPUT)
C                           APPLIED TO THIS CODE. NO OTHER WARRANTY,
C                           EXPRESSED OR IMPLIED, IS APPLICABLE.
C
      SUBROUTINE USPKD  (PACKED,NCHARS,UNPAKD,NCHMTB)
C                                  SPECIFICATIONS FOR ARGUMENTS
      INTEGER            NC,NCHARS,NCHMTB
C
      LOGICAL*1          UNPAKD(1),PACKED(1),LBYTE,LBLANK
      INTEGER*2          IBYTE,IBLANK
      EQUIVALENCE (LBYTE,IBYTE)
      DATA               LBLANK /1H /
      DATA               IBYTE /1H /
      DATA               IBLANK /1H /
C                                  INITIALIZE NCHMTB
      NCHMTB = 0
C                                  RETURN IF NCHARS IS LE ZERO
      IF(NCHARS.LE.0) RETURN
C                                  SET NC=NUMBER OF CHARS TO BE DECODED
      NC = MIN0 (129,NCHARS)
      NWORDS = NC*4
      J = 1
      DO 110 I = 1,NWORDS,4
      UNPAKD(I) = PACKED(J)
      UNPAKD(I+1) = LBLANK
      UNPAKD(I+2) = LBLANK
      UNPAKD(I+3) = LBLANK
```

```
  110 J = J+1
C                                    CHECK UNPAKD ARRAY AND SET NCHMTB
C                                    BASED ON TRAILING BLANKS FOUND
      DO 200 N = 1,NWORDS,4
         NN = NWORDS - N - 2
         IBYTE = UNPAKD(NN)
         IF(IBYTE .NE. IBLANK) GO TO 210
  200 CONTINUE
  210 NCHMTB = (NN + 3) / 4
      RETURN
      END
C
C
C   ================================================================
C   =  SUBROUTINE DATA - SUPPLIES THE INITIAL GUESSES WITH A FILE  =
C   ================================================================
C
      SUBROUTINE DATA(X,N,N1,NE1)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(N)
C
C PREDEFINED INITIAL GUESS
      X( 1)=  1.1
      X( 2)=  0.9
      X( 3)= -1.1
      X( 4)=  0.9
      X( 5)=  1.1
      X( 6)= -0.9
      X( 7)= -1.1
      X( 8)= -0.9
      X(9 )=  0.45
      X(10)=  0.55
      X(11)= -1.8
      X(12)=  2.2
      X(13)=  0.45
      X(14)= -0.55
      X(15)= -1.8
      X(16)= -2.2
C
      X(17)=  0.225
      X(18)=  0.275
      X(19)= -2.8
      X(20)=  3.2
      X(21)=  0.225
      X(22)= -0.275
      X(23)= -2.8
      X(24)= -3.2
```

```
      RETURN                                                                      FR011C50
      END                                                                         FR011C60
```

APPENDIX C

DATA SIGNAL PLOTS

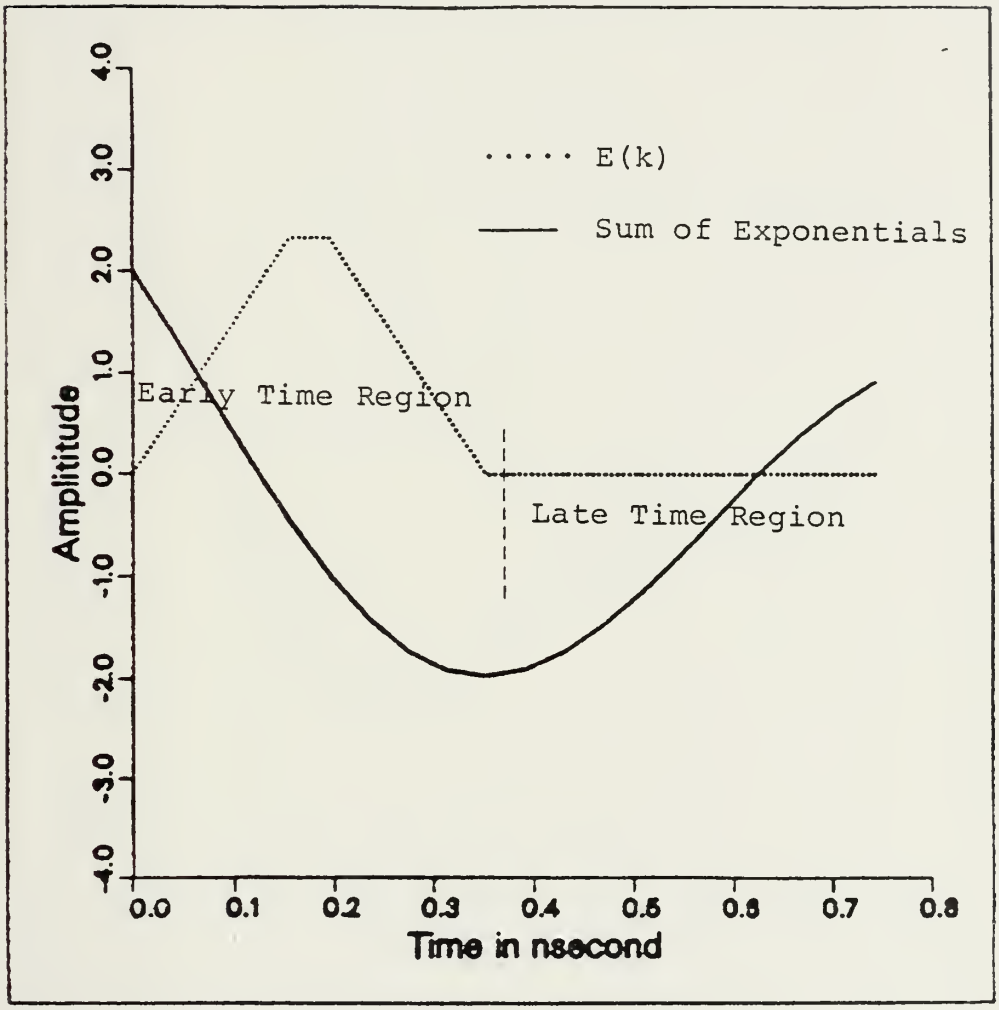

Figure C.1. Decomposition of Signal 1

(Noise Free)

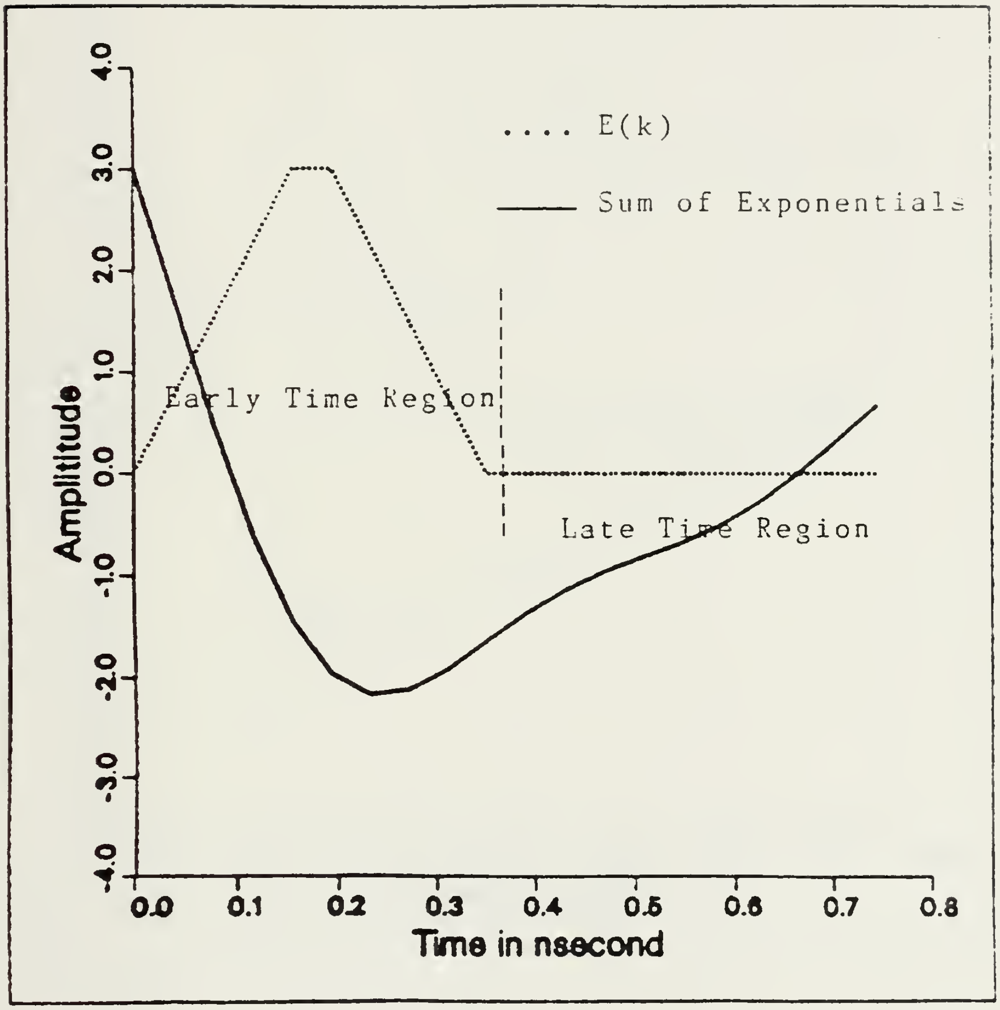

Figure C.2. Decomposition of Signal 1

(Noise Free)

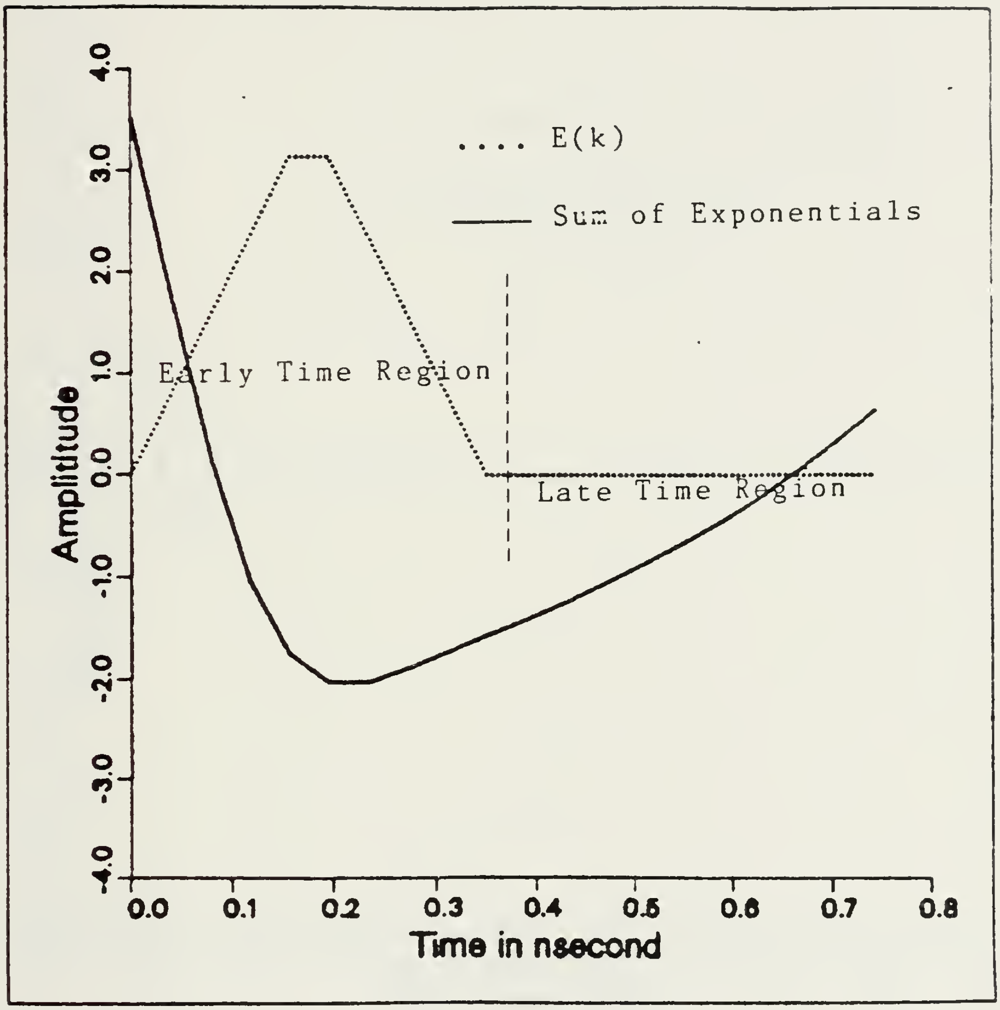

Figure C.3. Decomposition of Signal 3

(Noise Free)

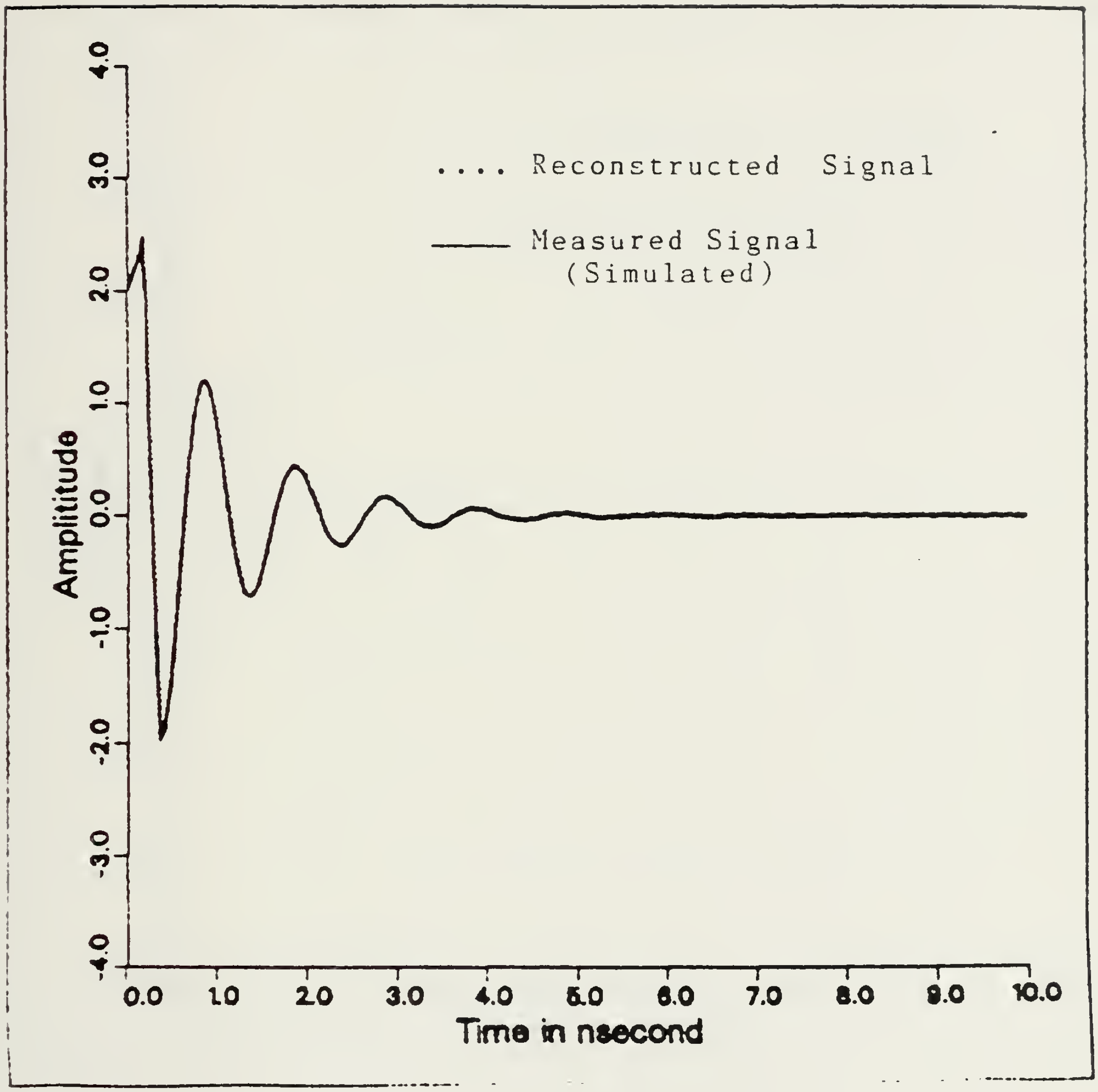

Figure C.4. Reconstruction of Signal 1 from The Computed Poles and Residues(SNR=30db)

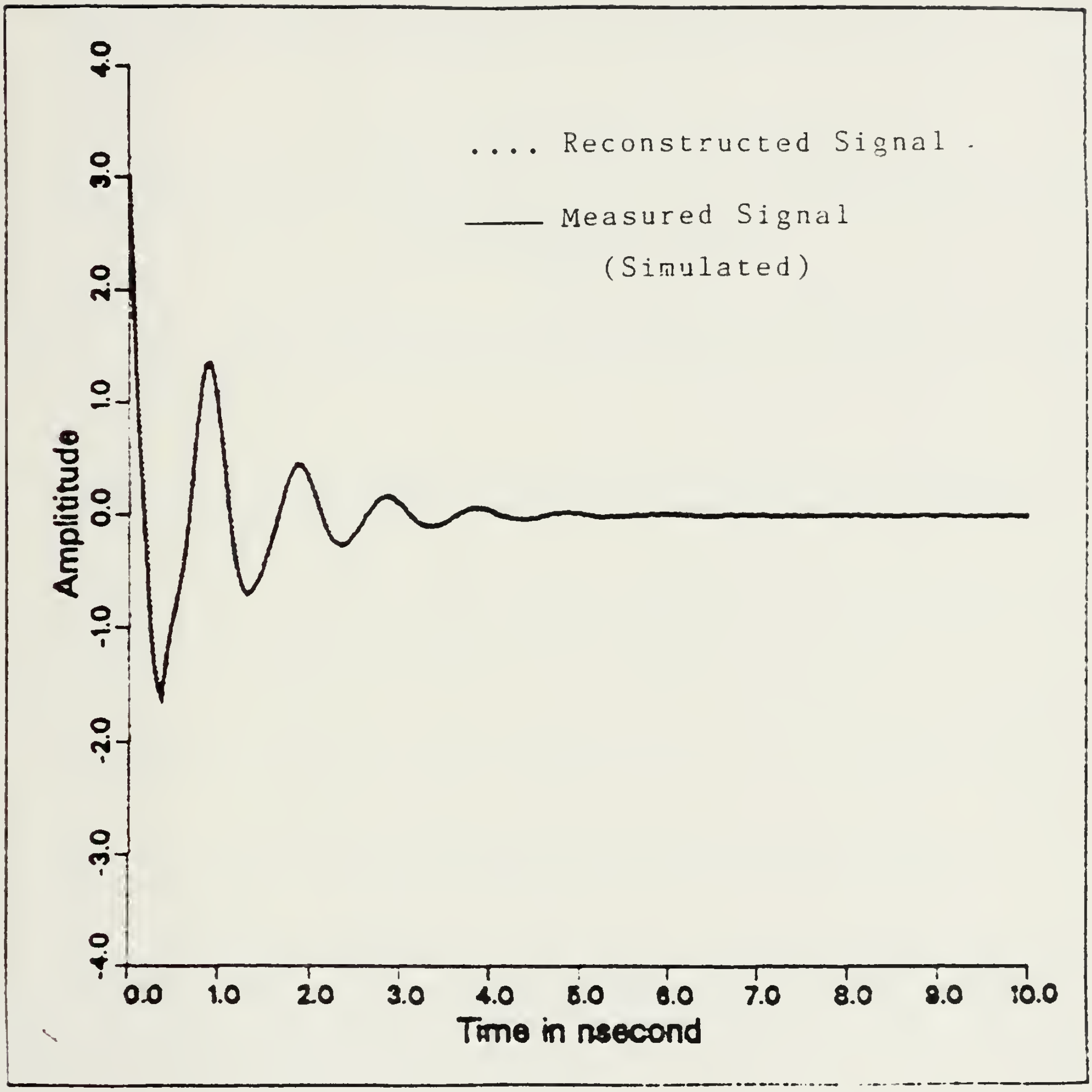

Figure C.5. Reconstruction of Signal 2 from The Computed Poles and Residues(SNR=30db)

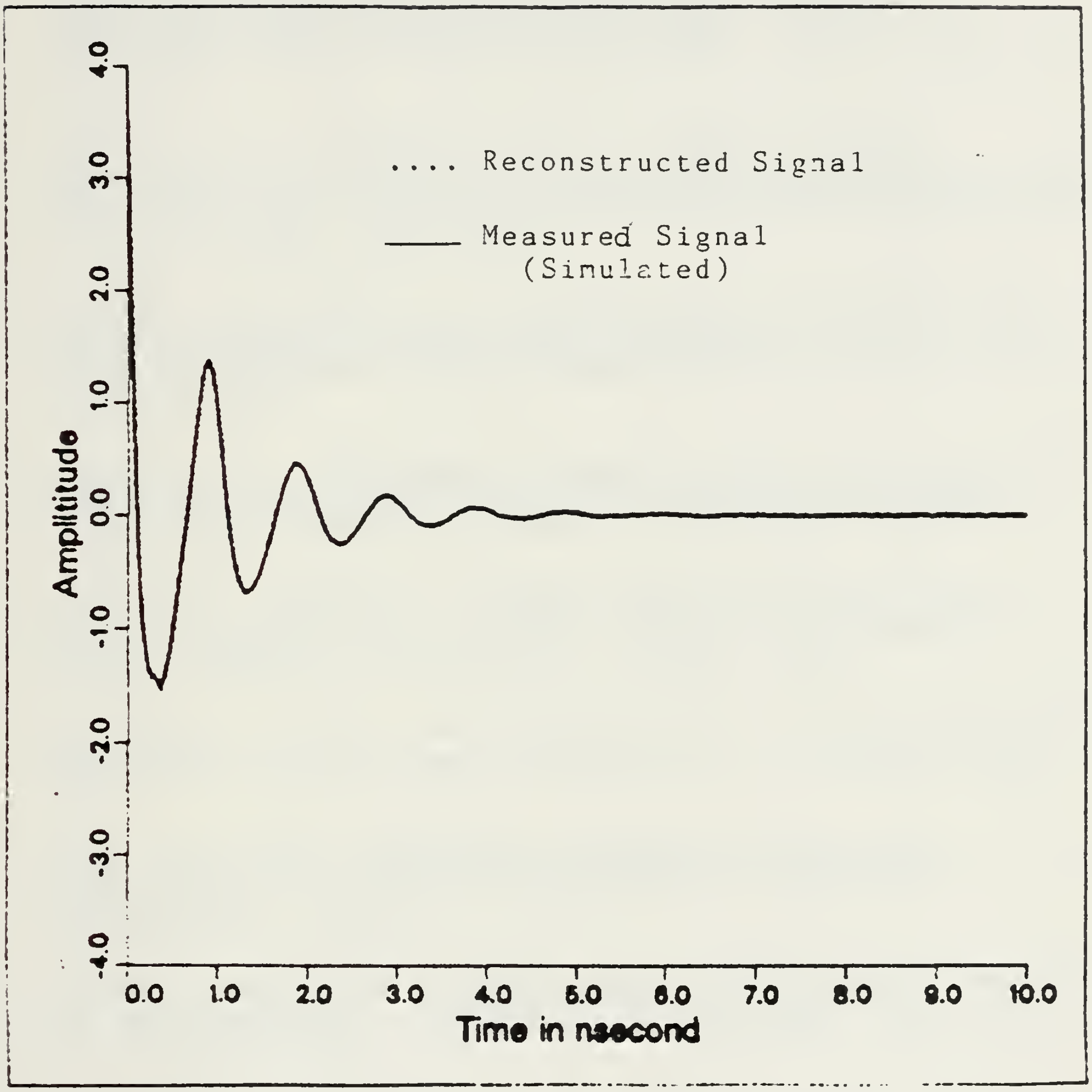

Figure C.6. Reconstruction of Signal 3 from The Computed Poles and Residues(SNR=30db)

LIST OF REFERENCES

1. Baum, C.E., Singularity Expansion of Electromagnetic Fields and Potentials Radiated From Antennas of Scattered From Objects in Free Space, Sensor and Simulation Notes, Note 179, Air Force Weapon Laboratory, May 1973.

2. Mains, R.H., and Moffatt, D.L., Complex Natural Resonances of an Object in Detection and Discrimination, Ohio State University, Electro Science Laboratory, Columbus, Ohio, June 1974.

3. Manilha, Fernando M.C., Investigation of Method for Natural Response Target Identification, Master Thesis, Naval Postgraduate School, Monterey, California, December 1982.

4. Morgan, M.A., Singularity Expansion Representation of Fields and Currents in Transient Scattering, Naval Postgraduate School, Monterey, California, 1983.

5. Davenport, Richard W., Target Identification Using Prony's Method, Master Thesis, Naval Postgraduate School, Monterey, California, October 1982.

6. Marquardt, D., H., "An Algorithm For the Least-Squares Estimation of Non-Linear Parameters", J. SIAM, 11(2), 1963.

7. Bard, Yonathan, Non-Linear Parameter Estimation, International Business Corporation, Cambridge, Massachusetts, 1974.

8. Bennett, Leonard C., "Time-Domain Electromagnetics and Its Applications", Proceeding of the IEEE, Vol. 66, No. 3, March 1978.

9. Levenberg, K., "A Method for the Solution of Certain Non-Linear Problems in Least Squares", QUART. APPL. MATH 2, 1944.

10. Van Blarucum, M.L., and Mittra, R., Techniques for Extracting the Complex Resonances of a System Directly From Its Transient Response, AFWL-TR-76-250, University of Illinois, Electromagnetics Laboratory, Department of Electrical Engineering, Urbana, Illinois, April 1977.

11. Prony, R., Essai Experimental et Analytique sur les Lois de la Dilatibilitie des Fluides Élastiques et sur Celles de la Force Expansive de la Vapeur de I'AIKOOI a Differentes Temperatures, J. I'Ecole Polytech. (Paris), Vol. 1, No. 2, pp. 24-76, 1975.

12. Householder, A.S., On Prony's Method of Fitting Exponential Decay Curves and Multiple-Hit Survival Curves, Oak Ridge National Laboratory, Report ORNL-455, Oak Ridge, Tennessee, February 1950.

13. The IMSL Library Volume 4 - International Mathematical & Statistical Libraries Inc., Houston, Texas, June 1982.

14. DISSPLA - Display Integrated Software System and Plotting Language Pocket Guide, Integrated Software Systems Corporation, San Diego, California, 1981.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943	2
3.	Department Chairman, Code 62 Department of Electrical Engineering Naval Postgraduate School Monterey, California 93943	1
4.	Professor M. A. Morgan, Code 62Mw Department of Electrical Engineering Naval Postgraduate School Monterey, California 93943	10
5.	Professor Hung-Mou Lee,Code 62Lm Department of Electrical Engineering Naval Postgraduate School Monterey, California 93943	1
6.	LCDR Choong Yeon Chong 100-33, Kyo-Dong, Chun-Cheon SI, Kangwondo, Seoul, Korea	3
7.	Division Chief Education Division Department of Personnel and Administration R.O.K. Navy Headquarters Seoul, Korea	1

205817

Thesis
C448872 Chong
c.1
Investigation of non-
linear estimation of
natural resonances in
target indentification.

17 JAN 89
11 NOV 89
7 FEB 90

80019
35,57,33,9

Thesis
C44887
c.1
205817

Thesis
C448872 Chong
c.1
Investigation of non-
linear estimation of
natural resonances in
target indentification.
AMERICAN BINDERY
TOPEKA KANSAS

thesC448872
Investigation of non-linear estimation o
3 2768 001 02657 8
DUDLEY KNOX LIBRARY